

Amazing Computing™

March 1986
Vol. 1 / Number 2
U.S.A. \$2.50
Canada \$3.50

Commodore Amiga™ Information and Programs



The Excitement Continues.....

Your first Amiga Basic program, **SuperTerm**

✓ Tutorials,
CLI Ed
Lisp
Amazing "C"

✓ Reviews,
Live!
Online!

✓ AMICUS

✓ Public Domain
Software

✓ Regular columns:
Miga Mania
Roomies

✓ Amiga Products



Amazing Computing & Amazing Dealers

The following are **Amazing Dealers**, dedicated to supporting as well as selling the Commodore-Amiga™. They carry **Amazing Computing™**, your resource for information on the Amiga™.

If you are not an **Amazing Dealer**, but would like to become one, contact:

PiM Publications
P.O. Box 869
Fall River, MA. 02722
1-617-679-3109

Alabama

Abax Data Systems
Madison Books & Computers
Mel's Photo & Computer Shop
Command Computers

Huntsville
Madison
Montgomery
Birmingham

Arizona

Computer West
Copperstate Business Sys.

Phoenix
Phoenix

Arkansas

SIS Inc.
The Micro Shop

Mt. Home
Little Rock

California

Home Computing Center
Computerland Santa Rosa
ComputerLand of Stockton
Brown Knows Computing
Computer Nook, Inc.
The Computer Room
Computer Attic
HT Electronics

San Bruno
Santa Rosa
Stockton
Redlands
San Bernadino
Scotts Valley
Palo Alto
Sunnyvale

Colorado

Computer Room
Micro World
Sun Country Computers
Ideal Computer Systems
Whole Life Distributors

Aurora
Lakewood
Colorado Springs
Evergreen
Englewood

Connecticut

Connecting Point

Georgetown

Spectrum Computers
Mnemonics Inc.

New Haven
Norwalk

Florida

Cotronics Inc.
Computer Terminal Inc.
Education Computers Etc.
Florida Book Store
Commodore Computer Shop
MSI Business System
Computer Bar
Micro's Etc.
Computer Image
Gulf Coast Computer Electronics
The PC Collection
New Age Electronics
Softwalls Computer Center
Orlando Electronics Co.
Computers Plus
Computer Ware Inc.
Family Computers
Random Access Computers
The Open Door

Stewart
Sebring
Tallahassee
Gainesville
Ocala
Brooksville
Pennsacola
Altamonte Springs
Miami
Panama City
Coral Gables
St. Petersburg
Holly Hill
Orlando
South Dayton
Tarmac
Sarasota
Ft. Walton Beach
Cocoa

Georgia

Future System
Software South Inc.

Augusta
Savannah

Illinois

Computer Resolution
Computerland O'Hare
BA Business Computers

Champaign
Park Ridge
Batavia

| | | |
|--|---|---|
| <u>Indiana</u> Greg Chaney Burkat Computer Ctr. Vons Computers Computer Corner | Greenfield South Bend West Lafayette Fort Wayne | Computer Outlet Computers Etc. Jamestown Syracuse |
| <u>Kansas</u> MidKansas Computers Data Bank Corp. Midwest Computers | Newton Wichita Manhattan | <u>North Carolina</u> Digitz TDS Computers Raleigh Carrboro |
| <u>Louisiana</u> Software Center Int. The Computer Clinic Software Mart | Metairie Lafayette Metairie | <u>North Dakota</u> Computer Associates Fargo |
| <u>Maine</u> Valley Computers | Auburn | <u>Ohio</u> North Coast Programming Quality Computer Appliances Quality Computer Appliances Softwaire Centre Int. Earthrise Micro Systems Saxtan Computer Center Computers Plus of Ohio Software and More Computer Network Lakes Consumer Electronics North Coast Programming Solon Toledo Toledo North Olmsted Columbus Sandusky Upper Sandusky Cincinnati Newark Akron Willoughby |
| <u>Maryland</u> Computer Crafters The Logical Choice Inc. Delta Gama Corp. | Wheaton Baltimore Gaithersboro | <u>Oklahoma</u> Video-Comp Inc. Colonial Video & Computer Second Hand Software Lawton Bartlesville Oklahoma City |
| <u>Massachusetts</u> The Bit Bucket Tycom Inc. OmniTech Computers Computerland HCS Computer Center LCA Video & Computer Ctr. Memory Location | West Newton Pittsfield Tewksbury Leominster Marshfield Norwood Walpole | <u>Oregon</u> Coackamas Computers IB Computers Computron Business Sys. Computron Business Sys. Coackamas Portland Portland Tigard |
| <u>Michigan</u> Midland Computer Shop Specialists Computer Roseville Computer Store Michigan Software Bits-Bytes-Nibbles Slip Disk Programs Unlimited Chelsea Computer Edmunton Computer Center Slip Disk Galaxy Computers Retail Computer Center Inc. The Software House | Midland Livonia Roseville Novi Petowskey Madison Heights Grand Rapids Ann Arbor Pontiac Clarkston Allen Park Farmington Hills Kalamazoo | <u>Pennsylvania</u> Basic Computer Systems Pittsburgh Computer Store Data Softique Computerware Hermitage Pittsburgh Pittsburgh |
| <u>Minnesota</u> MCD of Hibbing Inc. OnLine Computers Computer Specialties Plus Computers, etc. Computer Place | Hibbing Minnetonka Monticello Eagan Minneapolis | <u>Rhode Island</u> Micro Limits Micro Limits CompUtopia CompUtopia International Computer Ser. Smithfield Warwick Warwick Providence Johnston |
| <u>Missouri</u> Systems Plus Brands Mart Computers | Independence Kansas City | <u>South Carolina</u> Avcom International Inc. Horizon Horizon West Columbia Greenville Greenville |
| <u>Nebraska</u> BULLS-EYE Double E Electronics | Lincoln Omaha | <u>Tennessee</u> Software First Nashville |
| <u>Nevada</u> Century 23 Inc. ComputerLand of Carson | Las Vegas Carson City | <u>Texas</u> The Computer Experience Software Terminal Regency Educational Sys. Computer Time Inc. Softex Micro Search Inc. Computer Age Computer Concepts Softwaire Centre Computer Age Byte by Byte San Antonio Fort Worth Dallas Marshall Watauga Houston Houston Waco Dallas Dallas Austin |
| <u>New Jersey</u> Family Computing Centres Family Computing Centres Family Computing Centres | Fairfield Ridgewood South Orange | <u>Utah</u> Computers Plus Murray |
| <u>New Mexico</u> New Horizon Computer Sys. Page One Newstand Academy Computers | Alamogordo Albuquerque Albuquerque | <u>Virginia</u> Virginia Micro Systems Woodbridge |
| <u>New York</u> Software & Such Software Supermarket Byte Shop Ray Supply Inc. Castle Computer Tedron Business Products Star Tech Systems Amuse Computer Club World Computers Inc. Leigh's Computers Video Computer Center | Scotia Kenmore Merrick Glen Falls Latham Rochester Massena New York Hicksville New York Rome | <u>Washington</u> A.P.P.L.E. CO-OP Micro Age Computer Store Computron Business Sys. Renton Bellview Yakima |
| | | <u>Wisconsin</u> TMW Software Inc. Colotron Computers Software Shoppe Wausan Racine Fond Du Lac |
| | | <u>Distributed in Canada by</u> Avita Software Distributors and Phase 4 Distributors Inc. 1-800-661-8358 Concord, Ontario Southeast Calgary, Alberta |

MetaScope: The Debugger

MetaScope gives you everything you've always wanted in an application program debugger:

- **Memory Windows**
Move through memory, display data or disassembled code, freeze to preserve display and allow restoration.
- **Other Windows**
Status windows show register contents and program state with freeze and restore; symbol, hunk, and breakpoint windows list current definitions.
- **Execution Control**
Breakpoints with repetition counts and conditional expressions; trace for all instructions or subroutine level, both single-step and continuous execution.
- **Full Symbolic Capability**
Read symbols from files, define new ones, use anywhere.

- **Powerful Expression Evaluation**
Use extended operator set including relationals, all assembler number formats.
- **Direct to Memory Assembler**
Enter instruction statements for direct conversion to code in memory.
- **and More!**
Log file for operations and displays, modify/search/fill memory, etc.

MetaScribe: The Editor

MetaScribe has the features you need in a program editor:

- **Full Mouse Support**
Use for text selection, command menus, scrolling — or use key equivalents when more convenient.
- **Multiple Undo**
Undo all commands, one at a time, to level limited only by available memory.
- **Sophisticated Search/Replace**
Regular expressions, forward/backward, full file or marked block.
- **Multiple Windows**
Work with different files or different portions of the same file at one time.
- **Keystroke Macros**
Record keystroke sequences or predefine, assign to keys you choose.
- **and More!**
Copy between files, block copy/move/delete, set tabs and margins, etc.

Metadigm, Inc.

Metadigm products are designed to fully utilize the capabilities of the Amiga™ in helping you develop your programs. If you're programming the Amiga, you can't afford to be without them.

Dealer Inquiries Welcome

MetaScope
\$95.00
MetaScribe
\$85.00
MetaTools
\$69.95

(California residents +6%).
Visa/MasterCard accepted.

Amiga is a trademark of Commodore-Amiga Inc.

19762 MacArthur Blvd.
Suite 300
Irvine, CA 92715
(714) 955-2555

Dedicated to the memory of

Gregory B. Jarvis
Christa McAuliffe
Ronald E. McNair
Ellison S. Onizuka
Judith A. Resnik
Francis R. Scobee
Michael J. Smith

and to
all Pioneers who have paid the
highest of prices to extend our
dreams and frontiers.

We are poorer by their loss, yet
richer by their dreams.

Amazing Computing

Publisher: Joyce A. Hicks
Circulation Manager: Doris Gamble
Assistant to The Publisher: Robert James Hicks
Corporate Advisor: Robert H. Gamble

Managing Editor: Don Hicks
Hardware Editor: Ernest P. Viveiros
AMICUS Editor: John Foust

Amazing Authors:

John Foust
Don Hicks
Kelly Kauffman
Perry Kivolowitz
Richard Miner
George Musser Jr.
Daniel Zigmund
&
The Amigo

Special Thanks to:
Robert H. Bergwall
RESCO, Inc.
David Street
E.P.V. Consulting
New England Technical Services
Interactive Tutorials Inc.

Advertising Sales:
1-617-679-3109

Amazing Computing™ (ISSN 0886-9480) is published by PIM PUBLICATIONS, Inc., P.O. Box 869, Fall River, MA. 02722. Subscriptions: in the U.S., 12 issues for \$24.00; Canada and Mexico, \$30.00; Overseas, \$35.00. Printed in the U.S.A. Copyright © 1986 by Pim Publications, Inc. All rights reserved.

Amazing Computing

Table of Contents

March 1986

| | |
|--|--|
| Electronic Arts Comes Through Electronic Arts' first Amiga Software packages are reviewed | 7 |
| Inside CLI: part two George Musser's continued guide to CLI focuses on ED | 9 |
| Summary of ED Commands | 13 |
| Live! Richard Miner reviews a beta version of Live, the frame grabber | 14 |
| Online! and the CTS Fabrite 2424 ADH Modem John Foust takes a close look at two telecommunications products produced and distributed by Micro-Systems Software, Inc. | 17 |
| "ROOMERS" The Amigo returns with more "behind the scenes" information | 21 |
| The Amazing... C Tutorial: Part Two John Foust continues his investigation of C | 24 |
| The Amazing Lisp Tutorial: Part Two Daniel Zigmond returns to the versatile Lisp | 31 |
| Commodore's Great Educational Advantage Commodore has sent the Amiga to college | 34 |
| Amiga Products A listing of available and promised Amiga hardware and software | 36 |
| SuperTerm V 1.0 Kelly Kauffman begins a look at Amiga Basic with a "New and Improved" terminal program | 42 |
| Miga Mania Perry Kivolowitz returns to question and inform | 49 |
| The AMICUS Network: Amicus A Workbench "MORE" Program Amiga BBS Numbers Public Domain Programs, a listing of Amazing Computing Disks | 54 by John Foust 58 by Rich Wirch 61 62 |
| Departments: From the Editor Letters Index of Advertisers | 4 6 64 |

From the Editor:

by Don Hicks

Thank You. The response we have received from you, the Amiga users, has been wonderful. We have received calls from people all over the United States and Canada who have heard of Amazing Computing™ from either a Bulletin Board System, a friend or their dealer. The excitement these users generate is more than contagious, it is extraordinary.

We have even received calls thanking us for publishing Amazing Computing™. Imagine, calls from people excited about the same things that we are excited about and thanking us for doing something we are having so much fun doing. Ain't America Great!

When we first started Amazing Computing™ we were really looking for a source of information we could use on our Amigas, when we found so many fellow Amiga owners looking for the same source of information, we knew we had to do something. We were on Compuserve, asking some of the contributors if they would like to write for a "newsletter" on the Amiga. The response was tremendous. We finished with our premiere issue.

But, that was not the end, only the beginning.

This month we are continuing the tutorials that we started in our first issue and have added several reviews. And we have "graduated" from our dot matrix look to a lazer printer typeset.

Before you begin to feel that we are just getting pretty, consider that now we can offer more information in a nicer, easier to read package on the same amount of pages. This means that we have added quantity and quality without raising our basic production costs.

We are expanding in more directions than just printing, we are expanding the Amicus Network section to include "more" programs and articles on developing for the Amiga. We want a good strong development team of authors to continue to produce better ideas and broaden the base of knowledgeable Amiga programmers.

Sound strange? How better to increase the success of this magazine, the dealer who sold it to you and the machine you bought it for, then to produce more people who are excited and knowledgeable to program our favorite computer?

If you feel that you have a good idea for an article or a different way of approaching an old program, drop us a line. John Foust is looking for real hard core development style articles for the Amicus section.

I am a little easier. I am looking for articles that teach, entertain and take the mystery out of our Amigas, yet keep the fun in using our machines. If you take pride in being a little different (remember, if you own an Amiga, you have taken the road of an independent thinker), then you have the skills to expand our present frontiers.

Please, do not be shy. We will work with writers who have more imagination and creativity than English skills. If you have information that you want to tell your fellow Amiga users, let us help you and ourselves extend our boundaries.

One of the more difficult parts of this job is to stop writing and send the issue to the printer. There are two very good excuses for this.

One, with the stream of Amiga information increasing daily, it is very hard to finish the issue and route to the printer. A case in point is the fantastic digitized image disk I received from Tim Jension at DIGI-VIEW just now. These pictures are fantastic!

Two, we all suffer from the "hacker" syndrome. No, this is not the current popular definition, but the original definition. Hacker, a person who is never satisfied with what he or she has written (ie. as in program or prose) and is in a constant state of revision, where said writing never is completed.

In the second instance, we have a great deal of company in the Amiga community. Almost every programmer I speak with tells me that they are almost finished with this fantastic product, they just need to put a few more "touches" on the program and/or the documentation

The greatest thing we can do, is get on with it!



Don Hicks
Managing Editor

New Amiga Products From The Developers of Amiga C.

Amiga C Compiler—\$149.95
Everything you need to develop programs on the Amiga, including a full set of libraries, header files, an object module disassembler, and sample C programs.

Unicalc—\$79.95 A complete spreadsheet package for Amiga, with the powerful features made popular by programs such as VisiCalc, SuperCalc, and Lotus 1-2-3. Unicalc provides many display options and generates printed reports in a variety of formats and print image files. Supports 8192 rows of 256 columns, and includes complete on-line help.

Lattice MacLibrary—\$100.00
The Lattice MacLibrary is a collection of more than sixty C functions enabling you to rapidly convert your Macintosh programs to run on the Amiga. This allows you to quickly and efficiently take advantage of the powerful capabilities of the Amiga.

Lattice Make Utility—\$125.00
Automated product generation utility for Amiga, similar to UNIX Make, LMK rebuilds complex programs with a single command. Specify the relationships of the pieces, and automatically rebuild your system the same way every time.

Text Utilities—\$75.00 Eight software tools for managing text files. *GREP* searches for specified character strings; *DIFF* compares files; *EXTRACT* creates a list of files to be extracted from the current directory; *BUILD* creates new files from a batch list; *WC* displays a character count and a checksum of a specified file; *ED* is a line editor which utilizes output from other Text Utilities; *SPLAT* is a search and replace function; and *FILES* lists, copies, erases or removes files or entire directory structures.

Lattice Screen Editor (LSE)—\$100.00 Fast, flexible and easy to learn editor designed specifically for programmers. LSE's multi-window environment provides the editor functions such as block moves, pattern searches, and "cut and paste". Plus programmer features such as an error tracking mode and three assembly language input modes.

OTHER AMIGA PRODUCTS AVAILABLE FROM LATTICE:

Panel: Screen Layout Utilities—\$195.00

Cross Compiler:

MS-DOS to Amiga C—\$250.00

dBC III:

library of data base functions—\$150.00

Cross Reference Generator—\$45.00

With Lattice products you get *Lattice Service* including telephone support, notice of new products and enhancements, and a money-back guarantee. Corporate license agreements available.



Lattice

Phone (312) 858-7950 TWX 910-291-2190

INTERNATIONAL SALES OFFICES:

Benelux: De Vooght. Phone (32)-2-720-91-28. England: Roundhill. Phone (0672) 54675

Japan: Lifeboat Inc. Phone (03) 293-4711 France: SFL. Phone (1) 46-66-11-55

The Amazing Mail:

Dear Amazing,

If you bring all you propose in the Premiere Issue to print, it will certainly be "amazing". Also, of great benefit to the dummies like me with the new Amigas.

Looking forward to a very close and comforting relationship. Everything looks good, keep up the high expectations.

Sincerely,
Don J. Reed
Salisbury, MD

Dear Amazing Computing:

I have found your first issue to be most pleasing and informative. The information that you are providing would be hard to come by unless one spent all their waking hours scanning BBS's. Thanks for putting it all in one place. Keep up the good work.

Sincerely,
Phillip V. Petroska
No. Andover, MA

Dear Editor;

Congratulations on your Premiere Issue of Amazing Computing! I read your magazine cover-to-cover nonstop. It's been a long time since any magazine has so totally captured my interest. The very first issue is worth the subscription price to me. I had caught the Date Virus! You gave me the cure. I like your present format just fine. I would like to offer some advise as to the future format if I may. Please, please if you must use those awful tear-out advertisement post cards, put the ad on the left page and begin each new article on the right page. That will allow the cards or their remaining parts to act as indices to the articles. Have you ever noticed that if you try to fan through a magazine containing those cards that you only get to see those pages?

I am probably not the only one to point this one slight error in the C Tutorial by John Foust. On page 41, the output of the printf() statement in the fourth paragraph will not give the desired results as indicated because

the variables apples and bananas have been listed in the wrong order following the text to print.

Keep up the good work and once again Congratulations on a fine start.

Sincerely,
William H. Grosse
Greenville, AL

Gentlemen:

I have just finished reading your Premiere Issue, and am very pleased with its contents. I was particularly pleased with its initial orientation: covering hardware and higher level languages in addition to Basic. As more software becomes available from Commodore and third party developers, I look forward also to your coverage of the best of these.

Keep up the good Work.

Sincerely,
George H. Borrmann, Jr.
Schenectady, NY

Dear Sirs:

Ever since I bought my Amiga, I have been searching the news stands for literature dealing with the Amiga. There's not much out there, but in one issue (the first) your magazine presented more information than all of the other sources I have found put together. Thank you! Your magazine is in the class with the Amiga.

Sincerely,
Pat Jones
Michigan City, IN

Dear Sirs:

I enjoyed reading your Premiere Issue very much. Keep up the good work, and let's make the Amiga worth its weight in gold.

Sincerely,
John C. Ponte
East Providence, RI

Hello,

I am excited about your new publication - for two reasons - mainly because it proves that it can still happen in America (Entrepreneurality) and also because we need you (Amiga dedicates) in order to make use of our investment in Amiga. Our hats are off to ya'll here in the Carolinas, and based on what we see (even as computer novices) in "A MAZING COMPUTING", we are confident that our 24 bucks is well spent!!

Keep up the good work, and if we can be of assistance in promoting A/C & Amiga & AMICUS, just give us a hollar!

Best Regards,
Terry & Josi Krueger
Mauldin, SC

Dear Sirs:

Congratulations on being the first publishers with any decent AMIGA coverage. I have your Premiere Issue and I love it.

I'm already looking forward to your second issue. As a brand new AMIGA owner (1week) and a 3 1/2 year owner of an ATARI 400, I'm looking for technical information well beyond the level of "how a floppy disk stores data," or "how to care for your floppy disks". The tutorials look great. Other topics for tutorials might be: the 68000 instruction set, the relationship between Workbench and Amiga DOS (i.e. what files are necessary to have a useable Workbench with room for data storage), learning to think in "AMIGA" for experienced computer users. In other words lets get past the "any idiot can use it" stage and get close to our FRIEND.

I look forward to a long & happy association with your magazine.

Sincerely,
John Anderson
Regina Saskatchewan,
Canada



Electronic Arts Comes Through

by Don Hicks

We have seen the advertisements for months, "**Why Electronic Arts is committed to the Amiga**", with a picture of Electronic Arts president, Trip Hawkins, leaning on an Amiga. Scattered around the desk were the sphere, cube, and pyramid symbols of Electronic Arts while the Amiga displayed the same corporate images.

There was a quote from Trip Hawkins:

"The Amiga will revolutionize the home computer industry. It's the first home machine that has everything you want and need for all the major uses of a home computer, including entertainment, education and productivity. The software we're developing for the Amiga will blow your socks off. We think the Amiga, with its incomparable power, sound and graphics, will give Electronic Arts and the entire industry a very bright future."

The only exception we, at Amazing Computing, have with Mr. Hawkins' statement is the constant reference to the Amiga as a home computer. We believe this is only a small use of the Amiga's talents. However, Mr. Hawkins has proven true to his word on the quality of software Electronic Arts is preparing for the Amiga. Their Dr. J. and Larry Bird Go One-On-One and their Deluxe Paint did knock our socks off.

Dr. J. and Larry Bird Go One-On-One.

Ok, maybe it is because I am a Larry Bird fan that I enjoyed playing this game, but I don't think so. The best impression comes from the graphics and sound that Electronic Arts has generated on the Amiga as they "ported" this popular game from their other machine titles.

The game boots from the Workbench disk prompt. There is a bit of waiting while the game feeds in all the program and a rather long introductory music piece. You may listen to the music or press any key to go directly to the player menu.

Conveniently following the Amiga format, Electronic Arts has placed a menu at the top of the screen to set up each game. You have a choice of playing as Dr. J. or Larry Bird while the computer takes the other, or you can select the two player option and go at it with a friend.

In the two player option, one player must use a joy stick. The other player has the option of using a joy stick or the mouse. After using both, I do prefer the joy stick.

Under Options, you have a choice of either winner's outs or loser's outs.

The Skill menu gives you four choices of play: Park & Recreation, Varsity, College, and Pro. I saw no difference in the rules of the game, only the incredible increase of the speed.

The End game option allows you to end the game by time or points. Under Time you can set the clock for 2, 4, 6, or 8 minute quarters and under points the range is 1 point to 99.

Under the Game menu, you can resume your game, start a new one, or review the demo and music.

The action on the floor was good, the only thing lacking was the squeak of the basketball shoes across the parkay floor. Crowd sounds and voices were digitized into the program. Electronic Arts cleverly used the Amiga's multiple voice ability in layering sounds so there was never a drop in the excitement.

Penalties, calls and even an irrate janitor are all used expertly to add realism to the game.

The graphics are exciting with lots of color and action. We would have liked the characters of Dr. J. and Larry Bird to be a little less squat, but this is a very minor point to an excellent piece of programming.

Once you get the "feel" of the players movements, you become immersed in the action of the game. Occasional "instant replays" and "backboard smashing" gives the players a small break.

We were impressed with the game, although we have seen it on other computers, as well as with the manner Electronic Arts has mastered the Amiga.

Along with One-On-One, Electronic Arts has released several other great new additions.

Deluxe Paint. A graphics tool well worth the price. It gives a vision to your Amiga and with the new IFF standards, it is a great way to create images for your other applications.

(Editor note: We are working on a complete description of Deluxe Paint for our next issue. Yes, we did say we would look at it in this issue, but this is a good program and required a more intense look into its features and applications.)

Archon. A rather odd chess game for one or two players that is played on a board of 81 squares. Instead of the mundane pieces that we have used for years, we are treated to wizards, knights, dragons, shapeshifters, unicorns, goblins and more.

Each character has their own powers and weaknesses. You position them about the board and try to out maneuver your opponent. When characters do meet on the same square, they are removed from the board to a separate battleground. Here the game becomes an arcade style battle with the powers of the characters now thrown against each other.

Archon is an enjoyable game for the younger Amiga user in all of us.

The Seven Cities of Gold is an adventure for the adventurer in us all. How would you like to discover a new world. Or, if you choose, you can create another complete planet with new worlds and conquest aplenty. The game contains several modes of play with advanced degrees.

As any great explorer will tell you, you can die at this stuff. It takes a while to get comfortable with the controls and the strategy, but the game does open up new worlds to the Amiga explorer.

Financial Cookbook is not a game, unless managing your money seems like an imposible quest all its own. Financial Cookbook delivers a set of tools for projecting and managing your income in the "What If" stage.

Financial Cookbook is a direct port from earlier versions on other computers, however work has been performed to make this version correspond with the Amiga format. Financial Cookbook appears on the Workbench screen with its "dollar" icon. When clicked, it moves into a screen of gold lettering for the credits and then into the standard desktop with menu bar. From the menu bar, you choose the "recipe" (read program) you need for financial analysis.

Everything from leasing your car to long term investments is covered. If you have trouble attempting to refigure your finances, or simply want a small program to show you that you really could afford the Amiga, then the Financial Cookbook may fill the bill.

Electronic Arts has promised to produce new products and more games, we hope they do. They have taken the artistry of the Amiga and married it with the talents of their designers to create a set of programs that utilize the Amiga. Our only hope is that we see more programs specifically written for the Amiga. However, Electronic Arts has done what they are most talented at, producing good playable home computer games and products.

•AC•



Inside CLI: Part Two

by George Musser Jr.
CC004049@BROWNVN.BITNET

CIS 76566,3714

Opening up a CLI window peels back the Workbench and lets you peer into the underlying AmigaDOS. In fact, you can do things in CLI that you cannot from Workbench, such as access the RAM: disk, create a Startup-Sequence, and reASSIGN logical devices - topics we examined in the last issue.

This month, we will look into command files, investigate the AmigaDOS screen editor, and take a tour of the Workbench disk.

CLI itself is a fairly primitive environment. Basically, a CLI window just executes the filename typed by the user. Fortunately, AmigaDOS provides ways to tame CLI. One way is to create command files, also called batch files or execs. Command files contain sequences of commands for AmigaDOS to perform.

To invoke a command file, type the AmigaDOS instruction EXECUTE followed by the name of the file. You must specify EXECUTE; unlike MsDOS, AmigaDOS does not automatically recognize a command file. EXECUTE first looks in the current directory to find the file; if unsuccessful, it then goes to logical device S:, which is generally the s directory on the disk with which you booted-up.

The logical device S: can contain a special command file called Startup-Sequence, which the Amiga automatically EXECUTES when you boot-up. Here is a possible Startup-Sequence:

```
Echo "Workbench disk. Release 1.1"
Echo ""
Date < S:now
Echo "Please enter current time and date:"
Date > Nil: ?
Date > S:now
Echo ""
LoadWB
EndCLI > Nil:
```

This sequence of commands makes it easy to set the system date, if you have no battery-backed-up clock. The third line reads the file S:now to find when you last booted-up. The fifth line, **Date > Nil: ?**, asks the user for the new date, while the sixth line stores the new date in the file **S:now**. After the date has been set, the command file loads in Workbench and closes the CLI window. The Echo commands simply print the specified strings.

Command files can save you typing. For example, say you want to move files, that is, copy files to a new location and delete the original copies. The following command file will do the trick:

.Key OldName,NewName

```
; MOVE file to new location and delete original
FailAt 21
If >nil: Not Exists <OldName>
    Echo ""<OldName>"" does not exist."
    Quit
Else
    If Exists <NewName>
        Echo ""<NewName>"" already exists."
        Quit
    Else
        Copy <OldName> <NewName>
        Delete <OldName>
    EndIf
EndIf
```

The second line is simply a comment to remind you what the file does.

The first line reads the arguments and gives them the names OldName and NewName. The arguments are the filenames you type after the command, as in MOVE file1 file2.

In the third line, FailAt resets the lowest error code that forces the command file to abort. In this case, setting FailAt to 21 prevents the file from stopping if the user forgot the arguments. The next line checks whether OldName exists. If not, AmigaDOS tells the user and terminates EXECUTE. If so, AmigaDOS checks whether NewName exists; if it does, the command file halts, so that you do not accidentally overwrite a file.

Finally, if the parameters meet the tests, the command file COPYs to the NewName and DELETes the OldName.

Move illustrates many of the features of the advanced command files. Try creating your own simple command files. Such files may contain any valid CLI commands, a list of which were given in the last issue.

To create such files, you can call on the AmigaDOS screen editor, ED. ED is a nifty little editor you can also

use to write programs in C or BASIC programs and data files such as MaxiComm.FKeys.

To EDit a file, type ED followed by the filename and, optionally, the size of the file buffer, which is the memory where ED stores your file as you work on it.

ED <filename> (SIZE <n>)

The buffer defaults to 40k-bytes. If you find you cannot ED a file because you do not have enough memory, specify a smaller value of SIZE.

Once in ED, you can move around using the cursor keys, enter text, and invoke a variety of commands. There are two sorts of ED commands: immediate and extended.

You enter immediate commands by holding down the CTRL key and pressing the desired letter. For example, the CTRL-A combination inserts a line after the line where the cursor is; CTRL-B deletes the current line; CTRL-Y deletes characters to the right of the cursor; CTRL-D scrolls the screen down; CTRL-U scrolls the screen up.

To enter an extended command, press ESC. ED will

wait for you to enter the command and press RETURN. Extended commands include: B moves to the bottom of the file; F /string/ searches for a string; J joins the current line with the next; M <n> moves to line number n; Q quits the file without saving the text; SA saves the text; X saves the text and quits.

For a complete list of commands, see table one.

ED is hardly a word processor, but it does offer some nice features for text files, such as margins and word wrap, that is, when you type a word beyond the right margin, ED automatically moves that whole word down to the next line. You can set the left and right margins with the SL and SR extended commands.

The BS and BE commands select the start and end of a block of text, which you can delete, copy, and write to a file with the commands DB, IB, and WB respectively.

Of course, ED cannot format your text or give you fancy fonts, but it is compact, readily accessible, and free.

ED and the rest of the AmigaDOS commands live in logical device C:, which defaults to directory c of the Workbench disk. If you type the command ASSIGN without parameters, AmigaDOS tells you what logical devices, Input/Output devices, and disk volumes are currently defined. The ASSIGN command can add, change, or delete logical-device mappings, as described in part one.

Let's take a closer look at some of the AmigaDOS devices. Logical devices default to the directory with the same name on the boot disk.

SYS: stands for the SYStem disk. When you boot-up, SYS: maps to the root directory of the boot disk. Workbench looks in SYS: for the System directory, which contains the DiskCopy and Initialize tools.

The C: device contains CLI commands, which you can rename to match your preferences. In fact, you can stick any program you want into C: and AmigaDOS will treat it as a CLI command.

Operating-system libraries reside in L:. Ram-Handler manages the RAM: disk; it is loaded into memory the first time you copy a file to RAM:. From CLI, the RAM: disk looks just like a real disk drive. Unfortunately, you cannot see RAM: from Workbench releases 1.0 and 1.1. Another version of Ram-Handler, included with the pre-release Workbench 28.8 shipped to many dealers and developers, lets you access RAM: from the Workbench. This Ram-Handler works with the newer AmigaDOS and Workbench releases, leaving

Don't Be Shy! Communicate from your AMIGA

With ElTerm,

you can open your Amiga up to the world of telecommunications. ElTerm is a complete package for terminal emulation and file transfer. ElTerm can upload and download files using Kermit and Xmodem protocols.

ElTerm will be available first quarter 1986.
Suggested retail price is \$75.00 US
Dealer Inquiries Invited.

Elcom Software

16 Oak St. #2
New Brunswick, NJ 08901

Amiga is a trademark of Commodore-Amiga Inc.
ElTerm is a trademark of Elcom Software.

one to wonder why Commodore-Amiga shipped consumer machines without it.

In addition to Ram-Handler, L: contains Disk-Validator, which verifies the integrity of the disks you insert. It also resets the system date to the latest date of creation of a file on the disk. In AmigaDOS release 1.0, this could corrupt the dates of other files and disks, leading to a contagious date virus (see John Foust's "Date Virus", Amazing Computing, volume 1 number 1).

The S:, or Sequence, directory contains command files.

AmigaDOS searches the LIBS: device to find disk-based system libraries. The Amiga collects related routines into libraries. Some libraries are loaded from the Kickstart disk into write-protected memory and stay there until you turn off the machine. Other libraries are loaded from the LIBS: directory into read/write memory as needed; examples are the info.library, which manages the Workbench info command, and translator.library, which translates English text into phonemes for the speech synthesizer. When disk-based, transient libraries are no longer needed, they remain in RAM until a program needs the memory space they occupy.

Input/Output device drivers live in the DEVS: device. AmigaDOS loads them into RAM when you send data to a modem, a printer, or the speech-synthesizer. The clipboard.device takes care of the clipboard, used to exchange data among applications. The parallel.device and serial.device handle the parallel and serial ports. The narrator.device synthesizes speech. The printer.device basically filters the data according to the printer you select in Preferences and sends the data to either the serial or parallel port. The definitions of individual printers reside in the printers subdirectory of DEVS:.

FONTS: stores the optional text fonts. The Macintosh names its fonts after cities; Amiga's fonts bear the names of gems. You can view the font styles with the Notepad tool in the Utilities drawer of the Workbench disk. The Roman font used by Workbench and CLI is a topaz font found not in FONTS:, but in the ROM kernel contained on the Kickstart disk. Like libraries and I/O devices, fonts are loaded into memory when needed. However, an operating system bug can make the system think a font is in memory when it is not.

After the ASSIGN command lists the logical-device assignments, it shows the available physical devices, which might include DF0, the internal disk drive; DH0, the hard disk drive; PAR and SER, the parallel and serial ports; and CON and RAW, the system console.

You specify a CON or RAW device by describing the position and size of a window on the screen, as in:

CON:left/top/width/height/name.

CON and RAW differ in that CON filters out function keys and cursor keys and interprets certain control codes, whereas RAW sends the characters you type as-is. The screen output of CON and RAW look the same, except that CON intercepts a line feed and displays a carriage return/line feed sequence, which forces a new line. Word processors use RAW, while CLI uses CON.

Commands and programs run from CLI usually send their output to the CLI window. You can easily redirect input and output to different devices, and even to files. The greater-than sign redirects output; the less-than sign redirects input.

For example, consider the DATE command. Type DATE by itself, and the system time and date appear in the CLI window. The CLI window is the standard input and output device, given the symbol "".

"DATE > now" puts the date into a file called now.

"DATE > CON:50/50/300/100/Date" types the date in a window named Date.



YET ANOTHER UNFAIR ADVANTAGE.

Although you haven't had your Amiga for very long, you may find that you need a more powerful line interpreter. Consider these features:

- Pipes
- Search paths
- User definable command-line editing
- Definable function keys
- Unix-like wildcards
- More versatile redirections
- Command aliases
- Built in commands
- Command history

All available now, at a reasonable price, from

Z O X S O

THE AMIGA TOOLSMITHS.

P.O. Box 283, Lowell MA, 01853-0283 USA

Unix is a trademark of AT&T
Amiga is a trademark of Commodore Amiga Inc.

"DATE ?" prompts you to enter the date in the CLI window.

"DATE < now" takes the date from a file called now.

"DATE < CON:50/50/300/100/Date > NIL: ?" opens a window named Date, in which you type the new system date.

The NIL: refers to the so-called bit-bucket device, which throws away any output it gets.

You can redirect the output of each CLI command at your convenience. Many CLI commands also take devices as arguments. For example, NEWCLI <device> will open up a CLI window in the specified device, as in:

NEWCLI CON:0/0/450/150/CLI.

Unfortunately, it is harder to change to default size and position of the CLI window opened by clicking the CLI icon from Workbench. At the end of this article is a little ABasiC program that will do the trick.

CLI offers you a porthole into AmigaDOS. If you are used to working with mainframe operating systems like Unix and CMS or with microcomputer OSs such as CP/M or MSDOS, then you will feel right at home in CLI. If you are a fan of the desktop metaphors, you will find CLI eases many of your chores. •AC•

Haven't You Set Your *AMIGA'S* Time And Date Once Too Often?

Introducing

A - T I M E

*A clock/calendar card with battery back-up,
so you will never have to set the time and date
in your AMIGA, EVER AGAIN!*

- Plugs into the parallel port.
- A completely transparent printer port is provided, with total compatibility to all I/O operations.
- Battery back-up keeps the clock/calendar date valid on power down.
- Custom case with a footprint of only 2 1/4" x 7/8" x 3 1/4" (W x D x H) in standard *AMIGA* color.
- Leap year capability.
- A - T I M E package contains:
 - 1-A - T I M E clock/calendar module
 - 1-3.5" DS Utilities Disk
 - Operating instructions

PRICE \$49⁹⁵

AVAILABLE: NOW

Mail check to:

AKRON SYSTEMS DEVELOPMENT (ASD)
P. O. BOX 6408 (409) 833-2686
BEAUMONT, TEXAS 77705

include \$3.50 for shipping and handling
For MC/VISA orders call (409) 833-2686
AMIGA is a trademark of Commodore - Amiga inc.

```
100 '----- Patcher -----
110 '
120 '          by George Musser Jr.
130 '          5 January 1986
140 '
150 ' This little routine places a specified patch at the
      desired place
160 ' in a file, generating a file with the extension
      ".Patched".
170 '
180 ' Name of file to patch
190 fil$="ram:CLI"
200 ' Desired patch
210 patch$="CON:0/9/530/191/New CLI Window"
220 ' Location of patch, found by using TYPE OPT H
230 loc%=&h264
240 ' Open files
250 open "i",1,fil$
260 open "o",2,fil$+".Patched"
270 ' Loop through file
280 patched%=0
290 ptr%=1
300 while not eof(1)
310 ' Fill buffer
320 line input #1,buf$
330 ' Replace lost linefeed
340 ' N.B. Watch out if last character in file is line feed
350 if len(buf$)<255 and not eof(1) then
      buf$=buf$+chr$(10)
360 ' If already patched, then finish off file
370 if patched% goto 570
380 ' New position in file
390 ptr%=ptr%+len(buf$)
400 ' If not yet at patch site, dump buffer and try again
410 if ptr%<loc% goto 570
420 ' Output bytes before desired location
430 print #2,using "&";left$(buf$,len(buf$)-
      ptr%+loc%+1);
440 ' Output patch
450 print #2,using "&";patch$;
460 ' Make sure buffer is full
470 buf$=mid$(buf$,len(buf$)-ptr%+loc%+2)
480 while not eof(1) and len(buf$)<255
490 get #1,ch$
500 buf$=buf$+ch$
510 wend
520 ' Print remainder of buffer
530 buf$=mid$(buf$,len(patch$)+1)
540 ' Done with patching
550 patched%=-1
560 ' Put buffer in output file
570 print #2,using "&";buf$;
580 wend
590 ' Bye, bye
600 close
610 end
```

Summary Of ED Commands

IMMEDIATE COMMANDS

Cursor movement:

CTRL-D Scroll down half a screen
CTRL-U Scroll up half a screen
CTRL-E Move to top of screen, or to bottom if already at top
CTRL-I Move to next tab position
TAB
CTRL-R Move to end of previous word
CTRL-T Move to start of next word
CTRL-J Move to end of line, or to start if already at end

Text Insertion/Deletion:

CTRL-A Insert line after current line
CTRL-B Delete current line
CTRL-H Delete character to left of cursor
BACKSPACE
CTRL-O Delete word or spaces
CTRL-Y Erase to end of line

Miscellaneous:

CTRL-F Flip case and move cursor right
CTRL-G Repeat last extended command (see below)
CTRL-M Carriage return
RETURN
CTRL-V Redraw screen
CTRL-[] Enter extended command mode (see below)

ESC

EXTENDED COMMANDS

Begin with ESC and end with RETURN

Cursor Movement:

ESC-B Move to bottom of file
ESC-CE Move to end of line
ESC-CL Move left (nondestructive backspace)
ESC-CR Move right
ESC-CS Move to start of line
ESC-M n Move to line number n
ESC-N Move to start of next line
ESC-P Move to start of previous line
ESC-T Move to top of file

Text Insertion/Deletion:

ESC-A /string/ Insert line after current line
ESC-D Delete current line
ESC-DC Delete character under cursor
ESC-I /string/ Insert line before current line
ESC-U Undo changes made to current line

Block Manipulation:

ESC-BE Mark block end
ESC-BS Mark block start
ESC-DB Delete block
ESC-IB Insert copy of block
ESC-SB Show block on screen
ESC-WB Write block to file

Text Searches:

ESC-BF /string/ Find string, moving backwards (toward TOF)
ESC-E /str1/str2/ Find first occurrence of str1 and change to str2
ESC-EQ /str1/str2/ Find first occurrence of str1 and query whether to change
ESC-F /string/ Find string, moving forwards (toward EOF)
ESC-LC Consider case when searching
ESC-UC Ignore case when searching

File manipulation:

ESC-IF /filename/ Insert file
ESC-Q Quit without saving text
ESC-SA Save text
ESC-X Save text and exit

Margins:

ESC-EX Extend right margin (margin release)
ESC-SL n Set left margin
ESC-SR n Set right margin
ESC-ST n Set tab distance

Miscellaneous:

ESC-J Join current line with next
ESC-RP command Repeat command until error
ESC-S Split line at cursor
ESC-SH Show information on text



Live!

A Review
by Richard Miner

Live!, a new, real time video digitizer for the Amiga, is a great piece of hardware soon to be released by Commodore. For those of you who are into interactive video, image processing and analysis, computer graphics and art, or any related field, this product will surely be of interest. It is made by a small three person start-up company called A-SQUARED. I have been working with a beta-test version of the system for a few days and thought I would tell the world how Amazing it really is. I want to begin with a bit of terminology, then cover the hardware, software, and other features of the system.

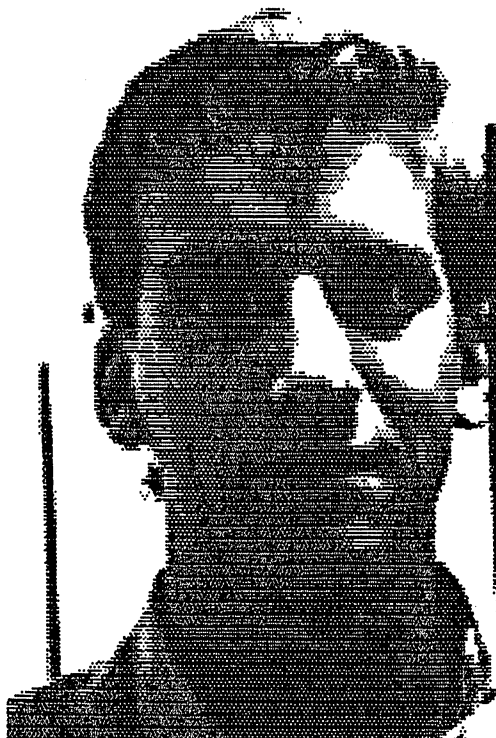
Let's examine what a digitizer does. A digitizer, or frame grabber, receives analog information, usually from a video camera. This information is actually a series of voltage levels representing, in an abstract form, the image transmitted from the camera. In order for the Amiga to understand and display this information, it must be transformed into a digital signal. This digital information represents the different brightness levels of the image being scanned. With Live! a black and white image from the camera will be converted into a picture with each picture element (pixel) being represented by one of 16 different levels of grey.

The system can display 16 levels of grey because there is an analog to digital converter (A to D) for the black and white component of the incoming RS-170 standard video signal. Each video signal is made up of

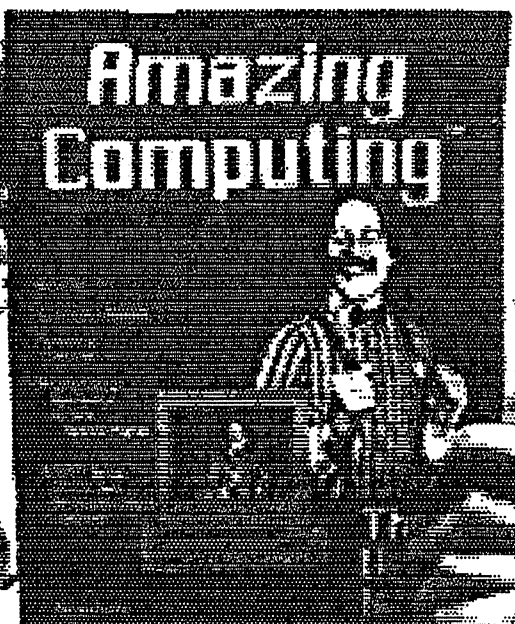
a series of scan lines, which represent horizontal slices of the complete video image. The A to D receives each scan line and samples it at intervals which correspond to individual pixels. Depending on the voltage level of the sample (ranging between zero and one) the A to D assigns one of 16 digital brightness levels to the scanned pixels. This is common in digitizers, but A-Squared went several steps farther. One of the special capabilities of Live! is that it can also digitize color images!

A-Squared built Live! so it would also digitize the standard for color images, NTSC. The Live! board uses an A to D converter to digitize the red, green and blue components of the incoming image in the same manner that the black and white signal was handled. Red green and blue are the primary colors of the video world, and are the three main components of color televisions. With four bits of information for red green and blue added to the 4 bits for black & white, the system has 12 bits for each pixel. This means the individual pixels which make up a digitized image can have one of 4096 different color values!

What do you do with 4096 colors? Good question, the Amiga can only display 32 at one time (in normal low resolution mode). Since all these colors cannot be simultaneously displayed, you must use software techniques to pick 32 colors for the Amiga. A good analogy to this is an artist who can mix only 32 colors to paint a given scene. You would need to look at the



**Richard Miner hard
at play with Live!**



**Mr. Miner with a friend as seen by LIVE! and
printed on an Epson printer (type unknown)**

scene you wanted to paint, and then mix the 32 colors getting enough of each different shade to paint the whole picture. This would in theory provide you with a relatively true color image on the Amiga. I say in theory since neither the folks at A-Squared nor us beta testers have done this yet, but it is a priority.

Live! got it's name because it can digitize an image so fast that the images can be displayed in "real-time". This means, as you move in front of the camera, your digitized image moves just as fast on the display. The lightning fast results are achieved because the Live! board acts as a direct memory access (DMA) device. The Amiga can transfer information to and from DMA devices without intervention from the CPU. With the Live! board sitting right on the Amiga buss, complete images of over 250 thousand bits (320 x 200 pixels x 4 bits for color) can be read into the Amiga's display memory in fractions of a second. The expansion buss is located on the right side of the Amigas case, and it is where you plug on the A-Squared Live! board.

As many of us know, it takes more than good hardware to sell a product such as this, you need good software also. Our beta version of Live! came with a mouse based, menu driven software package for "playing with" the features of the board. The basic features of the package are:
setting grey level intensities and ranges, adjusting image contrast through modification to the analog to digital conversion, changing color levels, assigning

color values to different grey levels, and saving an image.

The first two adjustments mentioned, basically allow you to modify the number of grey levels and their ranges. When you power up the software there are only two levels, black and white. This is because the initial values, for the version of the software that I have, only look at a narrow portion of the original video signal. You can change the parameters on the fly to examine a larger or more active portion of the original signal. For instance if most of the analog signal has voltages between 0.4 and 0.6, you could focus in on that part of the signal. This would result in the 16 different grey levels more accurately representing the scanned image. The image will look more like a black and white photo because you are telling the system to use more levels of grey in the incoming image signal.

The real fun begins when you start playing with the color controls. You assign different arbitrary colors to the various grey levels. Instead of black you might make it bright blue, and white might be red. And to each of the 14 levels of grey between black and white you could assign a different color. I assure you, the result does not look like the original image. Instead, it is a vibrant array of colors, all flowing as your scene moves. You can dance around in front of the camera for hours playing with different color combinations and modifying the hue and saturation of colors simply by rolling the mouse around. If you hook the system up to

INTRODUCING... THE MICRO-SYSTEMS PERSONAL PRODUCTIVITY SERIES

Analyze! is a powerful electronic spreadsheet program. Essentially, this program is a full-screen calculator where you can organize your data into rows and columns. These rows and columns can then be analyzed with simple mathematics or complicated formulas. Rows and columns can be duplicated to avoid re-typing. Both data and formulas can be edited with only a few keystrokes.

From home budgets and check registers to financial modeling and your company's general ledger, all manner of bookkeeping tasks become faster and easier with **Analyze!**. An astounding value at only \$99.95!

OnLine! combines features and convenience in a high quality package that will meet all of your telecommunications needs. With **OnLine!**, you can use your Amiga as a window to the world of information that is just on the other side of your telephone. You can link up with commercial information services for stock quotes, airline information and reservations, technical databases, and local bulletin board systems (BBS) and discover a new world of information and software for your computer. Corporate users can use **OnLine!** to let their Amiga access data stored on the company's mainframe computer.

OnLine! is the finest program of its type available for the Commodore Amiga. You can't lose when you get "online" with **OnLine!**. All for a down to earth price of only \$69.95!

Scribble! puts word processing efficiency at your fingertips. If you write anything from grocery lists to novels, you need this program. This is a word processor that helps you get your writing job done without needing a degree in physics to operate it. **Scribble!** is carefully designed with simple controls that make SENSE (unlike some other programs, which require you to take a week's vacation from work in order to learn how to use them). When you purchase **Scribble!**, you'll be using the program successfully only minutes later. Now THAT'S value for your dollar!

Despite its simplicity and whimsical name, **Scribble!** is a full featured word processor. Advanced formatting commands are included to give you a great deal of control over how your printed document will look. The editor will use any normal ASCII file (no rejection of carriage returns here!). It features multiple windows (up to four documents being edited at the same time) with block transfer between windows. And, through it all, **Scribble!** uses the famous Amiga Intuition interface. Menus are controlled with the mouse and all windows can be "sized" and moved as the user desires. **Scribble!** can even be printing out a document while you're using your Amiga for something else! Only \$69.95.

Micro-Systems Software, Inc.
4301-18 Oak Circle
Boca Raton, FL 33431

(800) 327-8724/National, (305) 391-5077/Florida

Ask about our Amiga bulletin board program, **BBS-PC!**. **AMIGA**, **OnLine!**, **Analyze!**, **Scribble!**, and **BBS-PC!** are trademarks of Commodore-Amiga, Inc., and Micro-Systems Software, Inc., respectively.

a VCR (via GenLock) you can be producing MTV quality videos in no time.

I mentioned before that you could save an image to disk, this is even better than it sounds! The images are saved using Electronics Arts IFF format. If you have not heard of this yet, you will in the future. Virtually all systems that read and write graphics images on the Amiga will be using the IFF format. What does this have to do with **Live!** you ask? Well for those of you with **GraphiCraft**, **Deluxe Paint**, or many of the other graphics packages, you can load in your digitized images and have a ball. The applications here are unbounded. I had fun drawing and writing captions on images, changing the colors, blotting out or adding details. You can start with a simple digitized image and end with colorful pictures and cartoons that you have personalized with the paint brush of your favorite graphics package.

You probably realize by now that we are a bit enthusiastic about this product, and have been having fun playing with it. We do plan on using **Live!** for more productive projects. These may include software systems for: image processing and image analysis as applied to machine vision applications, computer aided instruction, combined computer graphics and image manipulation software. We are open to suggestions and would like to hear what **YOU** would want in the way of software for such a fine enhancement to the Amiga. If you have ideas or applications in mind, please feel free to drop us a note at the address below, or to me in care of the **Amazing Computing™**.

It should be noted that our only affiliation with **A-Squared** is as beta-testers for their hardware. This review has been a simple description of the **A-Squared Live!** digitizer, and an accounting of our first impressions. If you want to pick one up at your local dealer, you will have to wait at least several months. I have been quoted second quarter '86 as a release date, with a price tag of around \$300.

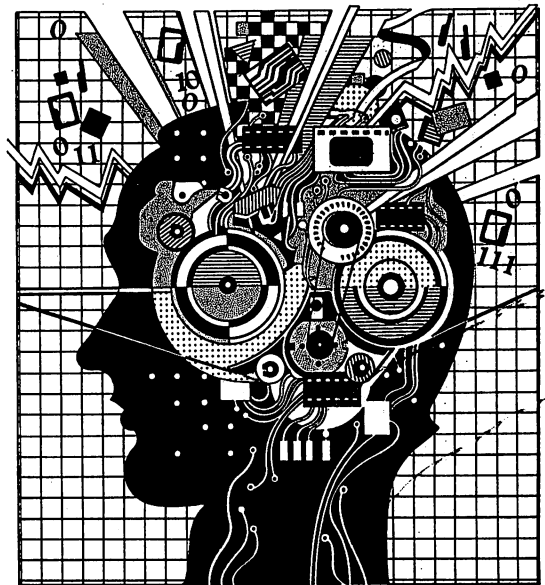
About the Author: Richard Miner is a student and researcher in computer science at the University of Lowell. His main interests are realism for computer graphics, standards for computer graphics, and robotics. He is also a tri-founder of **Zoxso**, a software developers consortium housed in Lowell. He can be reached at the following address:

US Mail: **Zoxso**
Usenet: ...!decvax!wanginst!ulowell!miner
PO Box 283
Lowell, MA 01853-0283

•AC•

Online! communications software and CTS Fabrite 2424 ADH modem

A Review by
John Foust



Online!

If you are a telecommunications junkie, a new terminal program can become your closest friend or a despised enemy. Every feature and every extra keystroke is tallied in your mind, comparing it to your last favorite program.

Online! was born at Micro-Systems Software, Boca Raton, Florida, in IBM's backyard. MSS also produces **Analyze!**, an Amiga spreadsheet, and has announced **Scribble!**, a word processor, and a package to turn an Amiga into a bulletin board system.

Telecom users have come to expect features such as terminal emulation, several file transfer protocols, and programmable function keys. **Online!** has these, and much more. It has an integral script language for automating login sequences and mail collection.

This feature places it above other Amiga terminal programs on the market today. I expect **Online!** will be the standard of comparison for many months to come.

At \$69, it is an unparalleled bargain, especially since MSS will undoubtedly offer upgrades in the future. **Online!** is a better product than most, by several orders of magnitude, and **Online!** works.

The **Online!** manual is a small, bound book of 105 pages, including a twenty-five page appendix describing the script language. It contains many relevant, easy-to-use examples, and is written in a clear style that anticipates many common questions. For example, it includes a description of necessary RS-232 cable pins. It does not contain an index, only a five-page table of contents. The disk is not copy-protected.

Automated login

My last telecom program was the public domain Kermit, for the IBM PC. It offered a simple command line interface, programmable keys, some online help, and the ability to remember the last four screens of information. Although this program is for a different computer, it has features that make it a joy to use.

For example, I subscribe to a half-dozen services, each with a different login sequence. In Kermit, I defined a series of function keys to the parts of the login sequence. One key dialed the number, and I press the others in sequence, F2-F3-F4-F5, and I am logged in. Mechanical, but it works.

One of **Online!**'s most powerful features is a script facility that can automate this process. With a few keystrokes, and your supervision, **Online!** will memorize a sample login session. It will create a script that will login automatically next time. No more function key sequences for me.

If you want to get your fingers dirty, you can write more complicated scripts to automate the retrieval of mail, too. Since the script includes commands such as 'WAIT UNTIL 12:30', your scripts can pickup your mail when network rates are low.

Using the examples in the manual, I quickly wrote scripts to login to three networks, all of which use different packet networks - TymeNet, Uninet and Telenet. Each worked the first time. I tried the automatic script generation method, and it worked just as well. The manual explained this process in detail. This is well integrated with the Amiga style. The

program lets you edit the text in question using the standard string gadgets.

To write my own scripts, I only had to learn two commands - WAIT and REPLY. My script looked like this for Compuserve, via TymeNet:

```
WAIT DELAY 10
REPLY "I"
WAIT STRING "Host Name:"
REPLY "CIS02|"
WAIT STRING "User ID:"
REPLY "72237,135|"
WAIT STRING "Password:"
REPLY "NOT.THIS|"
```

The vertical bar character represents RETURN in script strings, but the script syntax includes the ability to represent any character in a string, including control characters.

To login to Compuserve, I now type 'RUN ONLINE! CIS', and the program will login and start a transcript of my session, unattended. It would take only a few more lines in a script file to train it to get my mail. Online! has an icon, too, if you use the Workbench most often.

The script language includes keywords for IF, JUMP,

Amazing Writers

If you enjoy Amazing Computing and you are using your Amiga, you have completed one half of the qualifications of an Amazing Writer.

We are interested in the tasks and joys you have experienced on the Amiga. We want to read the secrets you have unlocked. We want to experience your excitement and enthusiasm. If you own an Amiga, you have already qualified as an Independent thinker, now use that ability to communicate your individual story or idea.

These pages are filled with people who wanted to reach you with their thoughts. They wanted to explain a portion of the computer you both use and abuse, because they thought it was interesting.

If there is something in the Amiga family that interests you, chances are there are people who would enjoy hearing what you have to say. So don't sit around waiting for others to teach you what you have already learned by hours of trial and error, get excited and teach the rest of us.

If your idea or explanation is of interest to developers and hard core hackers, please send your thoughts and a request for writer's guide lines to:

AMICUS Network Editor
PIM Publications
P.O. Box 869
Fall River, MA. 02722

If you are more interested in general use of the Amiga and it's products, please send your suggestions and ideas to:

Editor, Amazing Computing
PIM Publications
P.O. Box 869
Fall River, MA 02722

In both instances, please include your name, address, and phone number. We will return an answer as soon as our editors stop shouting about how great your idea is, and type a response.

WHEN (sort of like ON ERROR GOTO in BASIC), JUMP labels, colors, beeps, and ASK, a way of getting user input.

Definition files

In the same fashion, you can set all the parameters for your modem and each service you might call. Online! can redial until connection, and then execute the script. Each definition file presets all the available settings of Online!, and each file is only a pull-down menu away. These settings include the function keys, the file transfer protocol, and a link to the script and capture file names for each service.

File transfers

Online! can transfer files using Xmodem, Xmodem CRC, Compuserve B, or the Hayes Verification protocols. The Hayes protocol is used by the SmartCom package, the software sometimes bundled with Hayes modems. MSS has considered adding the Kermit protocol, a system popular among educational computer users. Kermit has the ability to transfer multiple files with a single request, a very handy feature when downloading many files.

While transferring a file, a small window appears to report the progress of the transfer. It has its own front-back gadget, so you could continue to work in some other program while you were transferring files. This is why I like the Amiga! On every other micro, I usually get up, make a sandwich, and wait, wait, wait during file transfers, since my machine is blocked from other uses.

Since I started the program with 'RUN ONLINE!', I can run other programs while online, up to the limit of memory. Online! uses 126,464 bytes of memory while running (without a capture buffer) and returns every byte to the operating system when 'Quit'. This is a sign of proper programming, and should be a test given in any review of an Amiga product.

The transfer window is nice, but only a block count is given during transfer. If you wonder how far along the transfer has come, you need to convert blocks to bytes, and then compare this to the known file size. I would rather see the decimal number of bytes transferred so far.

For a transfer from the Amiga to a host system, it would be possible to calculate a percentage, and display that, too. Online! only displays two block counts, total and transferred.

Also, a report of any errors encountered is absent from the transfer window, a common feature in most programs. There is room in the box to report the success or failure of the last several blocks transmitted, but it is not used. A report of this sort is necessary to evaluate phone line conditions, since line noise will cause errors in block transfer, and lengthen transmission time.

Although Online! does not have this feature, Amiga modem programs should be able to translate files before sending them with a protocol such as Xmodem. If you want to send a text file via Xmodem, chances are the host system prefers to see CR/LF at the end of each line. It could be possible to differentiate between sending Xmodem binary and sending Xmodem text, so that the LFs in Amiga files would be translated to CR/LF. Of course, one could use a filter in the CLI to pre-translate the file, and then send it.

Of course, as with any new product, it has a few misfeatures, but none are serious enough to discourage the purchase of Online!. In at least one respect, Online! does not follow the Amiga user interface philosophy.

Capture buffer woes

Like many telecom users, I prefer to keep a transcript of my entire session, and edit it afterwards, to remove the menus and junk messages. The cost of most services is too high to pause to read all messages while online. After all, you are paying for this, and you can not send all the text to the bit bucket with a clear conscience.

Online!'s capture ability uses a user-sizable RAM buffer, which can overflow to a disk file, a nice feature one would expect in a quality program.

During one session, I was capturing all of my work, and I wanted to send an ASCII file to the mail program on the network. Checking the manual, I learned the only way I could send an ASCII file was to close the capture buffer, (sending my session transcript to disk), and loading the letter I wanted to send, and then sending the capture buffer to the network.

After that, I would need to clear the buffer, and start the transcript again, with a different file name. This seemed an unnecessary hassle, especially since most terminal programs include a separate ASCII send and receive function, which does not affect the capture buffer. Online! should have a separate ASCII text send and receive function, and leave the capture buffer for capturing information.

Slow, expensive View

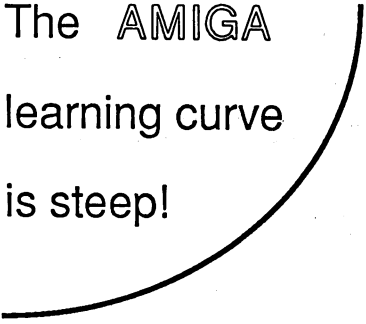
The best feature of a capture buffer is to look back at the last few pages that have now scrolled off the screen. In most online situations, a user encounters a menu, or a list of files, and then acts on that list. The action, as in downloading a file, usually scrolls enough so that the list is lost.

The only way to view the Online! capture buffer is sequential, from the start to the last character you typed. A new (resizable) window appears, and the capture buffer text is printed at a rate comparable to 1200 baud. If the capture buffer is more than a few pages long, this is useless, since the information you want is doubtless near the end of the buffer.

Meanwhile, the terminal mode is inactive, and there is no way to continue your work online while the capture buffer scrolls by. Of course, I discovered this on the most expensive network I use ...

In this respect, Online! could conform to Amiga user interface style. In this part of the program, and in the directory lists of scripts and archived transcripts, the standard Amiga scroll gadgets are nowhere to be found. If the capture buffer could be viewed with the traditional sliding bar, it would be a convenience instead of a booby trap. It should not prevent a person from working.

The AMIGA
learning curve
is steep!



Choose a porting house that's
well advanced along the curve!

Advanced Systems Design Group
Your Port of Entry into the Amiga Marketplace

280 River Rd., Suite 54A
Piscataway, N.J.
08854
1-201-271-4572

Amiga is a trademark of Commodore-Amiga, Inc.

Character mappings

Online! can map incoming and outgoing characters. It maintains seven separate one-for-one character translation tables for the display, the printer, the keyboard, and both to- and from- the capture buffer and communications port. For example, when I sent the letter to the network in the example above, I had to map the Amiga end-of-line character (LF) to the RETURN character (CR) the network requires. All these settings are saved in the definition file.

Online! can also handle the CR/LF translation in any fashion needed. a must for any Amiga terminal program.

Support and upgrades

MSS is active on Compuserve, since they are the SYSOPs of the TELECOMM SIG. They have been very responsive to questions and problems reported in the AMIGAForum, and via Easyplex. Upgrades to Online! can be obtained by modem, through the MSS BBS. They also offer technical support by telephone.

CTS Fabritek 2424 ADH modem

In the premiere issue of Amazing Computing, MSS presented a special offer - a 2400 baud Hayes-compatible external modem, a custom Amiga cable, and a copy of Online!, all for \$429. I had been searching for a fast modem, and this offer was too good to resist. (A 2400 Hayes modem normally costs \$500-700.) I used my credit card, and the modem arrived two days later via UPS.

The modem is made by CTS Fabritek, a company in Eden Prairie, Minnesota. It has a matte-finish plastic case the same color as the Amiga. It fits nicely on the top of an external drive, with a little room to spare. This is also a perfect size to place under a desk telephone. The two RJ-11 phone jacks come out the right hand side of the modem, towards the back. The external calculator-style power supply plugs into the back, along with a female DB-25 connector, and a positive action power switch.

The modem does not have an internal call-progress speaker, a definite minus for some users and applications. This monitor provides an extra assurance that a call is progressing normally. Otherwise, you have to keep your eyes on the screen, to see the modem's progress messages, or on the modem lights, to wait for the carrier detect.

It does not have transmit-data and receive-data LEDs, as some modems do. Again, these provide a certain level of assurance and evidence that your telecom program is working. During the testing of Online! and the modem, I sorely missed this feature.

It comes with a very complete guide to operation, more complete and useful than other modem manuals I have seen. For each setting or option described, it includes an example application where this feature might be used, and tips on troubleshooting related problems. I will keep this manual handy to help decipher other modems.

A true Hayes modem has two sets of DIP switches for setting modem operating parameters, such as always-active carrier detect, and other hardware settings. The Fabritek modem does not have DIP switches, but instead has an extra set of Hayes-like 'AT' commands, that all begin with 'AT&'. These software settings are stored in non-volatile RAM, which is nearly as permanent as DIP switches. The settings are not lost when the power is removed.

The extra 'AT' commands include the ability to store a single phone number in this memory.

I also used the modem on an IBM compatible, a Compaq Plus. At first, I tried to use a public domain background dialer program - one of those hacks to let a PC pretend it can multitask - to punch through to a local BBS. The program expected the carrier-detect line (CD) to go high when a connection was made, but the modem's default mode for CD is always high. A simple change with one of the custom 'AT' commands, and the default was reset and frozen into memory, and all went well.

I tested the 2400 baud speed on local and long-distance BBSs. I had no trouble calling the local BBS. In the course of three hours of transfers over three days, the modem was relatively free of glitches - no more than I had come to expect from a 1200 modem. I also called a BBS about 50 miles away. Again, the connection was as good as a 1200 baud modem. Both transfers went well - with only one or two errors.

However, during a long-distance call from Wisconsin to the MSS BBS in Florida, the 2400 baud carrier dropped once, but came back a few seconds later. The call was made on AT&T lines. In over a week of testing, the modem appeared robust, but only long-term use will prove that.

Online! has a list price of \$69. MSS sells Online!, the Fabritek 2400 baud modem, and custom Amiga cable for \$429. It is available from: **Micro-Systems Software**, 4301-18 Oak Circle, Boca Raton, Florida (305) 391-5077, National (800) 327 8724 •AC•

"Roomers"

by

The Amigo

When a computer that is different from the norm is first introduced, there is always a long wait for software and hardware to become available for it. Here it is in March. If you consider that the Amiga was shipped at the end of September and look at the many products that are being introduced for it, it surely says a lot for the new kid on the block.

This month, we are seeing many ways to upgrade your Amiga through hardware. Commodore-Amiga finished the expansion specifications in Mid-January, and have been working with third-party developers such as MicroBotics, Tecmar, CardCo, The Micro Forge, etc., to make sure the products will meet the interface specs. Commodore-Amiga informed some of its hardware developers to **change** their designs to meet this new expansion spec.

In theory, at least, the Amiga will automatically sense all your peripherals and configure them into the system for you ... including booting Kickstart and Workbench from your hard disk. In theory.

Tecmar, reacting to user feedback is reworking the T-disk so that the controller is housed inside the disk.

Meanwhile, **MicroBotics, Inc.**, is talking about shipping BIG hard disks next month. How big? How about your choice of 20, 40, 72 or 117MB? Suggested prices are \$1495, \$2159, \$3895 and \$7495, respectively. This Amigo has not personally seen them yet, but sources tell us these guys mean business! A 20MB to 40MB upgrade will cost \$695 if you want to go that route. Some plans for big memory expansion are in the works, but MicroBotics is going full out on the hard disks for now.

And for those of you who wish Amiga hard disks were as inexpensive as IBM PC hard disks, well... be on the lookout for a card for your Amiga that will allow you to grab any off-the-shelf IBM PC compatible disk drive

and slap it on the Amiga. Coming soon.

Also due this month is the framegrabber, now called **LIVE!**. It is supposed to knock your socks off. Beta testers say indeed it does. One great feature is that it saves images in IFF format, and DeluxePaint (or Graphicraft) will read them in, so you can edit them. Wow!

Hot on the heels of LIVE! is the **GenLock** device which will allow you to place a video image on the computer screen as the background color. Then you will be able to sync all this up to your VCR and blow everybody's minds.

Speaking of GenLock ... one recent demo went like this: a guy walks in with a video camera, fires up Boing! (the bouncing ball demo) and points the camera at other people in the store, telling them to jump up and down. They do, of course (what some people won't do for a demo), and it looks like the checkered ball is making these people go crazy!

Some of the people really got into it, pretending they were getting squished, etc. The whole thing was captured on VCR, courtesy of the Amiga's composite output. Now he takes an image and grabs it with LIVE!, saves it in IFF format, puts titles on it with GraphiCraft and prints a dozen copies of the screen for all around. Impressive. All anybody could say is "**what other machine could do this? None!**"

Cherry Lane Technologies has completed their music products. Cherry Lane is known for their professional music software and hardware, and they are really excited about the Amiga ... so you can imagine what they came up with!

The MIDI interface and the music keyboard are the first two products they have sold to Electronic Arts to market. Their other music products will be sold

through Commodore.

Software: It just never stops! A few announcements:

Lattice has recalled its MacLibrary. Seems that ALINK can not handle the large applications library without running out of memory (and you know what running out of memory is like!). Lattice is supposedly busy at work porting their linker to the Amiga. Advance reports claim it is smarter and faster than ALINK.

MicroSmith's 'TED' text editor has been renamed to 'TxEd' because of an existing product called TED. If you have not seen TED (oops, TxEd), you owe it to yourself. A real classy programmer's editor that takes full advantage of the Amiga's windowing capabilities.

And in other programming utilities, Zoxso has ZLI, a CLI replacement that remembers commands, loads function keys and more. Although I have seen announcements about it, I have not talked with any beta testers yet.

Due out this month: The PC Emulator. Newest inside reports say we will see about 75% the speed of a stock IBM PC. Price to be under \$200. Commodore is

promising an add-on board to provide the boost to make it as fast as an XT or so, but no real word on when that will happen.

MusicCraft will see the light this month. The pre-release version was plenty powerful, and Commodore claims that it is the 'beginner' package! What will they do for an encore (no pun intended)!

Developer warning: The Lattice C compiler and ALINK both can use lots of stack space when working with big files. You should edit your startup-sequence and put in a STACK 10000 command or something similar. I have STACK 20000 (just extra careful I guess) and have never run out of stack space during a compilation or link. Of course, reserving 10K for your stack reduces your working RAM by 10K...

More news from Lattice: Many developers are running V1.0 of Amiga C (really Lattice C V3.02). Lattice will upgrade you to V3.03 for a small fee, providing you can prove you bought the compiler or are a Amiga-certified developer. Meanwhile, Commodore is still trying to get the V3.03 disk out as Amiga C V1.1...

Watch out for a bug in V3.03. Instead of using:
`menu[i].nextmenu = &menu[i+1],`
use:

`menu[i].nextmenu = &menu[i]+1 ...`
Check your K&R to see why this works. If you have been scratching your head over why AmigaTerm will not compile, look for things such as: `&pointer[i+1]` and change them.

Other known bugs in 3.03:

1. `strtol()` is broken
2. `pow()` sometimes loses precision in the 13th or 14th decimal place
3. `fopen()` in append mode will instead truncate the file
4. expressions like `variable += (-1)` don't compile into correct code.

Lattice says that V3.04 will fix these bugs.

Speaking of bugs ... the Rom Kernel Manual states that library routines may only modify registers D0, D1, A0 and A1. But! Some routines are clobbering other registers! For example, `Text()` and `DoIO()` both clobber D6, and `DoIO()` sometimes (?) clobbers D7. So beware of the variables you bind to registers!

Batteries Included (from C64 fame) was pondering a port of DEGAS to the Amiga, but did not think anybody would pay \$40 for the program when Electronic Arts' Deluxe Paint was going to sell for \$65. Unfortunately, they were probably right.

HARVSOFT

PRESENTS

I N F O B A S E

Now available for the Amiga!

INFO BASE is a powerful, yet easy to use Data Base program, designed for the Amiga to store and retrieve information in an organized manner. Each record can contain up to 200 fields of information.

You create, design, and edit custom print forms. Each print form, you design, can create a different type of printout such as mailing labels, multi column reports, invoices, lists, etc.... In addition, text can be incorporated into the design of print forms, allowing both text and field information to be printed on the same line.

I N F O B A S E

FEATURES

- *DESIGN DATA BASES, ALLOWING UP TO 200 FIELDS OF INFORMATION FOR EACH RECORD
- *SORT RECORDS BY ANY FIELD
- *SELECT STARTING AND ENDING SORT INFORMATION
- *PRINT OUTPUT TO PRINTER OR SCREEN
- *PRINT MULTIPLE COPIES
- *CREATE, DESIGN, AND EDIT CUSTOM PRINT FORMS
- *COMBINE TEXT INTO THE DESIGN OF PRINT FORMS
- *ADD RECORDS AND DELETE RECORDS
- *BROWSE THROUGH RECORDS
- *SEARCH FOR SPECIFIC RECORDS

INFO BASE is available NOW!

\$45.00 (US)

Enclose Check, Money Order, or Cashiers Check.
Phone orders will be shipped C.O.D. and must add \$3.00 S & H.
Please allow 10-14 days for delivery.

Harvsoft

Box, 725
Kenmore, NY 14217
(716) 877-3510

Dealer Inquiries Welcome

INFO BASE requires an Amiga with 512K RAM and at least one disk drive.

A few assemblers to challenge MetaComco:

One is called DEVPAC, an assembler with built-in screen editor, and the other, called SEKA, is an assembler with debugger and disassembler. No word if you will ever see them, as both are from U.K. companies, and alas, neither has struck a U.S. distribution deal yet. We can only hope.

The last is from QUELO, which should be available any day now.

And in other MetaComco-related news, Commodore is trying to phase out ABasiC, now that MS-Basic is here. MetaComco reportedly does not want to see that, so look for a super-duper version of ABasiC with a compiler in the future.

The authors of BASIC have redesigned it and called it TrueBasic. They are porting it to the Amiga, and say it will be ready next month. I have heard doubts that they can make it that soon, but I hope they do.

In the Things That Should Be Written Department:

Last month I grumbled about the huge amount of terminal emulation software being written. This month, a suggestion: How about a font editor/creator? Wouldn't it be great to sit down and design a custom font for the Workbench screen, or any other screen? Everyone could donate his/her font to a public domain library (like AMICUS) for all to use. Let's get going!

Another idea: How about a menu system to create custom printer configuration files? These utilities are dying to be written!

Along those lines: There are hundreds of application programs just waiting to be written for the Amiga. What should they be? This Amigo personally knows of many developers that are eager and knowledgeable enough to deliver some powerful packages ... but do not know what to write!

Hey, you can help. Let's call it the **First Unofficially Sanctioned Amazing Questionnaire** (FUSAQ - don't you love acronyms?). We all have dreams of the ultimate software package for the Amiga. What is your dream? Send me some suggestions, I will publish a list. I bet the staff of Amazing would like to see some typesetting software!

Outta breath for this month. Keep those cards and letters coming!

•AC•

INTERACTIVE ANALYTIC NODE

THE EXPLORER

A debug monitor for the AMIGA

THE EXPLORER is a learning tool for new AMIGA owners and a debug tool for the program developer. THE EXPLORER is a machine language monitor similar to the ones you have used on 8-bit machines, but it has some design enhancements that make it more useful. Features: display memory and files in Hex and ASCII, memory modify, search, move, fill, display and change registers, disassembly trace, load programs, disassemble to disk. Output to printer or disk file. Powerful commands: loops, text display, real-time RAM view, & more!

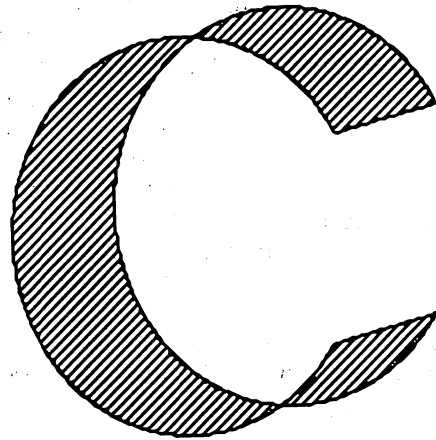
The EXPLORER doesn't force you to enter long commands to perform a function. The command set is compact and efficient. You can execute your commands within loops, creating live displays of RAM or registers while you test your program. You can control the display format too, and even display informative messages. No need to spend ten minutes typing that special data structure into memory every time you debug. Just write a new command to do it for you. After all, what are computers for? When you want to save the contents of RAM or a series of trace steps for future examination, just send them to the printer, or better yet, send them to a disk file!

PRICE \$35 plus \$3 S&H. COD add \$4. Visa/MC orders call (612) 871-6283. Money orders or checks to:

Interactive Analytic Node
2345 West Medicine Lake Drive
Minneapolis Minnesota 55441

The Amazing... C Tutorial, Part Two

by John Foust



My memories of tutorials recall a certain frustration. They were either too slow or too fast. If it was self-paced, I wanted to skip ahead, and get to the part where I could actually do something.

This phenomenon must be related to the tendency to assemble something before reading the instructions. You know you might miss something vitally important, but it is more fun to blaze ahead. When the lessons arrive only once a month, this tendency grows stronger.

I imagine several types of readers of this column: the people curious about the C language; people curious enough to buy a C compiler for the Amiga, but don't know C; people who hope to improve their C ability; and serious C programmers who hope to catch a mistake and write a letter to the editor.

This installment of this C tutorial should satisfy people in the first three classes, the people who want to get a simple C program up and running. It will introduce more C operators, the concept of a C expression, and flow-of-control statements.

In the first tutorial, a simple C function called 'mult()' was described. It will provide a good review of the basic structure of any C function. It returns an integer, the product of two integer arguments:

```
/* a function to multiply two numbers */  
  
int mult(a,b)  
int a,b;  
{  
    int c; /* a local variable */  
    c = a * b;  
    return(c);  
}
```

Recall the basic structure of any function in a C program: first, declare the result type of the function. If this is a function that multiplies two integers, it should return an integer value. Second, name the function, and include a left parenthesis.

If the function will be passed values, list the local names of these parameters, and then give the closing parenthesis. If the function had parameters, declare the type of each parameter, in order. Then the function itself can begin, with the left brace character.

The body of the function begins with this brace. Within the body of the function, local variables can be declared in the standard C fashion: first list the type of the variable (such as *char*, *int*, or *float*), the name of the variable, and a final semi-colon.

The C assignment operator '=' is obvious to anyone who has programming in an algebraic language like BASIC, Pascal, or FORTRAN. C has a wealth of other operators; some are obvious, some are more obscure. C has '+' and '-' for integer and floating point addition and subtraction, '*' for multiply and '/' for divide. '%' is the modulo operator, sometimes known as the remainder operator.

Here is C source code demonstrating these operators:

```
int a, b, d;  
  
a = 3;  
b = 9;  
d = a * b; /* d gets 27 */  
d = a / b; /* d gets 0 */  
d = b % a; /* d gets 0 */  
d = a << b; /* d gets 1536 */
```

A key concept in understanding C programs is the notion of **expression**. An expression can be an arithmetic expression, such as the lines above. C also has logical expressions, with corresponding logical operators.

An expression can be as simple as the name of a variable. The value of the expression 'apples' is the value stored in the variable 'apples'. The value of the expression 'apples+10' is the value stored in the variable 'apples', plus the value 10. This expression has a type of 'int', if 'apples' is an 'int'. C has several rules for the evaluation of expressions. We will continue to discuss expressions.

Instead of an arithmetic value, logical operators return values that represent 'true' and 'false'. A true logical expression always returns the value 1, while a false logical expression always returns 0.

These logical operators will look familiar to you. '>' is 'greater than', '<' is 'less than', '<=' is 'less than or equal to', and '!=' is 'not equal to'. While some versions of BASIC allow '<=' for 'less than or equal to', C does not. Logical 'equal' is '==', which differentiates it from '=', the assignment operator.

The '!' in '!=', or 'not equal to' is itself a C logical operator, called 'not'. It performs a logical compliment: if an expression is true, '!' makes it false. Other logical operators include '&&' for logical AND and '||' for logical OR.

C can easily manipulate the binary bits in a variable, too. The shift operators '<<' and '>>' move bits left and right in a value. A left shift fills with zeroes, a right shift may or may not fill with zero bits.

Does it seem strange that the definition of a programming language can be so imprecise? This example points at one of the problems with C. One version of C may fill with 0, one may fill with 1. Often, this is a result of the underlying machine language of the processor of the computer.

If a programmer writes a program that depends on the bits being filled with 1, the program will not work on a machine that fills shifted values with 0. This problem is referred to as 'portability'. These minute differences in C implementations hamper the transport of C programs from one type of computer to another.

C has other binary bit operators. '&' is AND, '|' is OR, '^' is XOR, and '~' (tilde) is NOT. (Don't fret if you don't know what these do.)

Here is more source code demonstrating these operators:

```
int a, b, d;

a = 3;
b = 9;
d = a << b; /* d gets 1536 */
d = a & b; /* d gets 1 */
d = a | b; /* d gets 11 */
d = ~a; /* d gets ??? */
/* it depends on the # of bits */
/* in an 'int' - this can vary */
```

Note that logical operators effectively return an arithmetic value, either 1 or 0. This value can be used in other arithmetic expressions:

```
int a, b, d;

a = 3;
b = 9;
d = (a > b) + 1; /* d gets 2 */
```

The parentheses in 'd = (a > b) + 1' guides the way the computer evaluates an expression. The expression within parentheses is evaluated first, as all algebraic languages do. The '(a > b)' is a logical expression that evaluates as true, so the arithmetic value of the expression is 1, and d gets 1 plus 1, or 2.

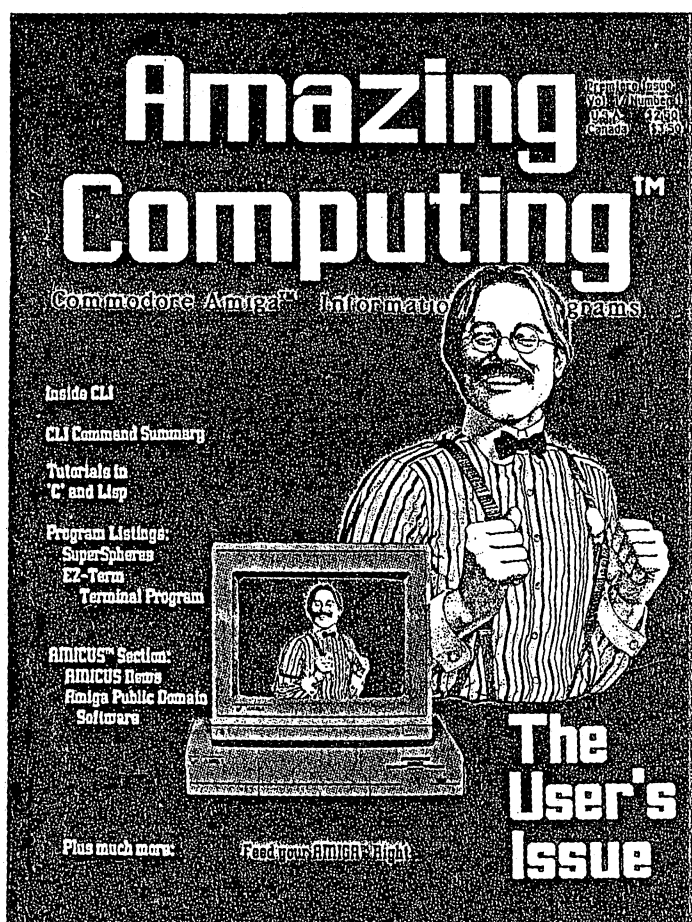
For example, the arithmetic expression '(b+3)*(d-9)' is composed of several smaller parts, '(b+3)' and '(d-9)', which are multiplied to give the expression value, -92.

Logical expressions can cascade in the same way arithmetic expressions can be composed of a series of arithmetic subexpressions. The logical expression '(a < b) > (b > d)' has two parts, '(a < b)' and '(b > d)'. Both are evaluated, and then the resulting true or false values are compared with the 'greater than' operator.

As Kernighan and Ritchie point out in "The C Programming Language", the expression '(a < b < c)' is not what it seems. First, the 'b < c' part of the expression will be evaluated, and then this true or false value will be compared to 'a'.

The notion of an expression is used throughout C. Functions can return values that can be used in other expressions, and the value of the whole expression can be assigned to a variable, or used to change the flow of control of the program.

A Classic



PiM Publications, Inc. wishes to thank all of the fantastic Amiga owners who have accepted Amazing Computing as a resource for their Commodore Amiga. The response was far beyond our expectations. Your letters praised Amazing Computing™ and encouraged us to continue to deliver Commodore Amiga information and programs.

Amazing Computing's success does have one draw back, our Premiere issue is in short supply. If you have not already purchased a copy of the Premiere issue from your Amiga dealer, or he has sold out, PiM Publications will make this back issue available for as long as they last at our standard Back issue price of \$3.00

Please send to:

**Back Issues
Amazing Computing
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722**

If you want to be sure you do not miss an issue, then subscribe!
Amazing Computing is available by subscription for 12 issues:

\$24.00 in the United States
\$30.00 in Canada and Mexico
\$35.00 Overseas

Amazing Computing™ your resource to the Commodore Amiga

Flow of program control in C is simple if you already know Pascal. If you are a die-hard BASIC programmer, it might take you a while. C has a 'goto' statement, but if anyone ever catches you using it, you will have a lot of explaining to do.

C has a rich set of control structures. The simplest is 'if':

```
if (a<b) {  
d = a - b;  
}
```

All statements within the braces will be executed if the value in 'a' is less than the value in 'b'.

If only one statement is present, then the braces are unnecessary, so:

```
if (a<b)  
d = a - b;
```

has the same effect as above.

'if' can be extended with 'else':

```
if (a<b) {  
d = b - a;  
}  
else {  
d = a - b;  
}
```

If 'b' is less than 'a', then 'd = a - b;' will be executed.

The braces can be left out, since only one statement is executed for each part of the conditional branch:

```
if (a<b)  
d = b - a;  
else  
d = a - b;
```

This points out the basic concept of a statement, as well as the idea that any set of statements enclosed by braces is executed as a single statement. A general syntax diagram for the 'if' construct is: 'if (logical expression) statement'.

This can be extended to include 'else': 'if (logical expression) statement else statement'. This type of syntax diagram is used to define many computer language grammars.

C has a counting looping construct, called 'for', similar to the 'FOR' of BASIC, Pascal, and FORTRAN. It is more flexible, however, as this example shows:

```
for (i=0; i<10; i=i+1) {  
d = a * b;  
}
```

The statements within braces will be executed ten times, with 'i' taking on the values 0 through 9. After the first parenthesis, and before the first semicolon, you may place a C statement that will be performed only once, as the loop is first started. If you want nothing to be initialized when the loop starts, just type the semicolon after the open parenthesis.

The middle statement, between semicolons, will be executed before each iteration. If it is false (zero), the loop is exited. Note that it may be a logical expression, or it may be a C statement, such as 'a = b'. When this expression is evaluated, and the result value placed in 'a', it is tested, and the loop may be exited based on this value.

The last expression, before the closing parenthesis, is performed after the loop statements are executed, and before the middle value is tested again.

In the last example, the variable 'i' is incremented by one. This operation is so common in programming tasks that C has special operators to do increments and decrements. The statement 'i=i+1' can be written as 'i++', with nearly the same effect. It can be written as '++i', too.

Within an expression or statement, these two have different effects. One is pre-increment, or '++i', which means, 'increment i before it is evaluated in this expression'. The other, 'i++', means 'increment i after it is evaluated in this expression'.

A different 'for' loop could look like:

```
for (d=0, i=0; i<10; i++) {  
d = a * b;  
}
```

Note that two statements are placed before the first semicolon. The two statements 'd=0' and 'i=0' are separated by commas. Both will be executed when the loop starts, so both 'd' and 'i' will be set to zero. Also, '++i' could be used as the last expression, with the same effect.

UBZ FORTH™

for the Amiga™

- *FORTH-83 compatible
- *32 bit stack
- *Multi-tasking
- *Separate headers
- *Full screen editor
- *Assembler
- *Amiga DOS support
- *Intuition support
- *ROM kernel support
- *Graphics and sound support
- *Complete documentation
- *Assembler source code included
- *Monthly newsletter

\$85

Shipping included
in continental U.S.
(Ga. residents add sales tax)

UBZ *Software*

(404)-948-4654

(call anytime)

or send check or money order to:

UBZ *Software*

395 St. Albans Court
Mableton, Ga 30059

Try FORTH Be Impressed

*Amiga is a trademark for
Commodore Computer. UBZ FORTH
is a trademark for UBZ Software.

The comma-separated statements in the last example could be part of an 'if' statement, too:

```
if (a<b)
d = 0, i = 0;
```

They would be treated as a single statement, and both would be executed if 'a' was less than 'b'. Of course, it would be clearer programming style to make these separate statements, and enclose them in braces.

Within an expression, the decrement and increment operators introduce useful effects. For example:

```
int i,j;

i = 3;
j = 4;
d = i++ * ++j;
```

means 'd' gets the value 15, and 'i' and 'j' are 4 and 5, respectively. The '++j' means 'j' is incremented before it is evaluated in the expression, so 'd' gets '3*5', or 15. 'i' is incremented **after** it is evaluated.

Here is another 'for' loop:

```
for (; i<10; ++i) {
d = a * b;
}
```

This does nothing on loop startup. The loop will execute a different number of times, based on the value of 'i' before the loop starts. If 'i' is greater than 9, then the loop statements between the braces will not be executed.

Again, the braces may be left out if only one statement is to be executed. Don't be perplexed by this example. The loop statement could be another loop statement. The construct is called a nested 'for' loop:

```
for (i=0; i<10; i++)
for (j=0; j<10; j++)
d = d * i * j;
```

Although this example makes little programming sense, since 'd' will get so many different values, this type of structure could be used to initialize an array of values to a constant.

This type of nesting can happen with the 'if' construct:

```
if (a<b)
if (c!=d)
  g = 3;
else
  g = 6;
```

The 'if (c!=d) g=3; else g=6;' is executed only if '(a<b)'. The 'else' does not refer to the 'if (a<b)'. Any dangling 'else' is matched to the most recent 'if'.

A consistent style of indentation is important in C programs. Of course, the compiler does not care about your style, but if you read this code in someone else's program, what would you think?

```
if (a<b)
if (c!=d)
  g = 3;
else
  g = 6;
```

At first glance, it would appear the author meant to make the 'g = 6' statement as the 'else' clause of the 'if (a<b)' statement. The compiler would not. It would bind the 'else' to the 'if (c!=d)'.

'while' is a construct similar to 'for'. A sample 'while' loop:

```
while (a<b) {
b = b - 2;
}
```

This piece of code would run until 'b' is smaller than 'a'. The expression between parentheses is evaluated before the statements between braces are executed, so if the expression is false to start with, the statements are never executed.

There exists a variant of 'while', called 'do-while':

```
do {
b = b - 2;
} while (a<b);
```

Where, the expression between parentheses is executed and tested at the end of the loop, after the statements have been executed. This construct guarantees that the statements between braces will be executed at least once.

It is similar to the Pascal 'repeat-until' construct, except that the 'do-while' statements are executed while the test expression is true, and the Pascal 'repeat-until' statements are executed while the test expression is false.

Sometimes a programmer uses the construct:

```
while (1) {
... some code...
}
```

Since the expression '1' will be true forever, the loop will be executed forever - an infinite loop. Usually, loops of this kind include a 'break' statement within the loop statements:

```
while (1) {
....
....
if (error)
  break;
....
....
}
```

Hey Amiga Developers!

Need a database capability in your application?

How about:

- o Arbitrary record and key construction.
- o Multiple indices freely intermixed.
- o Add, modify, or delete indices at any time.
- o Duplicate or non-duplicate key values.
- o Extensive caching of both key and data sets.
- o Multiple data sets concurrently.
- o Partial key searches.
- o Full error checking and reporting.
- o 16Mbyte data set capacity.
- o Full documentation.

Sources will compile and operate under UNIX Version 6, 7, System III, System V, 4.1BSD, 4.2BSD, 4.3BSD.

Manual cost will be applied to future purchase.

Dealer Inquiries Welcome

Advanced Systems Design Group

280 River Rd. Suite 54A Piscataway, N.J. 08854

1-201-271-4572

AMIGA is a trademark of Commodore-Amiga, Inc.
UNIX is a trademark of AT&T Bell Laboratories, Inc.

The 'break' jumps out of the loop, and executes the code after the closing brace of the 'while'.

In some programs, several different things may happen, based on the state of a particular value. This can be done with several layers of 'if' statements:

```
if (a==1)
b = 12;
else
if (a==3)
b = 8;
else
if (a==5)
b = 4;
else
b = 2;
```

This construct is usually indented only one level, for clarity. The same effect can be produced with a construct called 'switch':

```
switch (a) {
case 1:
b = 12;
break;
case 3:
b = 8;
break;
case 5:
b = 4;
break;
default:
b = 2;
}
```

If 'a' does not match 1, 3, or 5, then the code following the 'default:' would be executed. If the 'default:' and its code were removed, no code would be executed if 'a' did not match 1, 3, or 5.

The 'break' keyword jumps out of the 'switch' statement. If the 'break' was not present after each 'case : b = xx;', then the program flow would drop through the remaining 'case' statement groups, executing the 'b=8', 'b=4', and 'b=2' statements. Most often, the 'break' statement is used in this context.

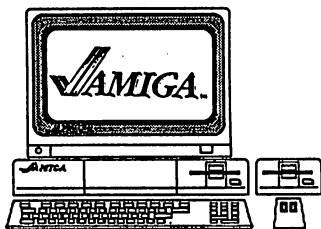
With these operators and control structures, plus the simple output function 'printf()', you can feel more confident about reading and writing C programs. One of the best ways to learn a language is to 'hear' it, so read and try to understand other examples of C programs in this magazine. There may be much you do not understand, but revel in the parts you do understand.

In our next tutorial, we will use the preprocessor, and declare other types of variables. Several more standard C input and output functions will be demonstrated, along with the Unix theory of command line arguments.

•AC•

Amazing Computing™

AMIGA™ Information and Programs



Feed your Amiga RIGHT! Don't miss an issue!!!

Charter Subscriptions

12 Informative Issues

\$24 United States
\$30 Canada and Mexico
\$35 Overseas

Send check or money order to:

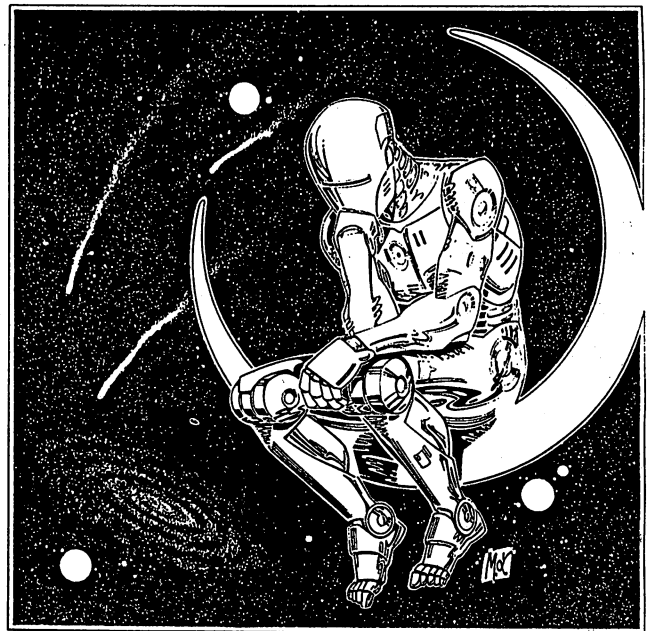
PiM PUBLICATIONS, Inc.

P.O. Box 869, Fall River, MA. 02722

Amiga™ is a trademark of Commodore-Amiga, Inc.

The Amazing Lisp Tutorial: Part Two

by Daniel Zigmond



In the last issue of **Amazing Computing™**, we explored how to manipulate both single units of data and lists of data. It is important to note that no matter how complex our symbolic expressions were, each one seemed to exist in a vacuum. For example, (car '(a b c)) would always return the value a, no matter what had been typed before it and it had no influence on the next expression we typed. Each line we typed after the "Input:" prompt was completely independent of every other line. In this installment of the tutorial, we will look at functions that have "side effects", that is, they change the values that other function calls will return.

Using only the math functions we have covered, we can find the square of 5 by typing (times 5 5). We can also find the square of the difference between 10 and 5 by typing (time (difference 10 5) (difference 10 5)). As you see, as the expression gets more and more complex, it also becomes more redundant. In this example, we always type the expression we want to square twice.

Lisp solves this problem by letting us store the values of expressions in "symbols". Symbols are similar to the variables of other languages except that they can hold any type of data (numbers, strings, lists, arrays, etc.). Every Lisp symbol has a name that is used to represent the value it holds. We can assign a value to a symbol with the function setq. For example, the following expression stores the value 10 in the symbol x:

```
Input: (setq x 10)
Value: 10
```

To retrieve the value of a symbol, we need only type it's name.

```
Input: x
Value: 10
Input: (setq a (difference 10 5))
Value: 5
Input: (times a a)
Value: 25
```

Now, we have solved our problem of redundant expressions. When we need to use the value of an expression more than once, we simply assign the value to a symbol and substitute the symbol for the expression.

There is one more point about symbols we need to examine before going any further. We can do this by looking back at one of our expressions from last month: (list 'a 'b):

```
Input: (list 'a 'b)
Value: (a,b)

Input: (list (plus 1 2))
Value: (3)

Input: (list '(plus 1 2))
Value: ((plus 1 2))

Input: (list 'times 2 3)
Value: (times 2 3)
```

Notice, when we preceded the symbols a and b with single quotes, they were not evaluated. The resulting list was (a b). This still holds true, even after we have given value to a. As long as there is a quote in front of a symbol, the symbol itself is returned and not its value. To make this more understandable, consider the following interaction:

```
Input: (setq a 10)
Value: 10
Input: (setq b 20)
Value: 20
Input: a
Value: 10
Input: 'a
Value: a
Input: (list 'a 'b)
Value: (a b)
Input: (list a b)
Value: (10 20)
```

At the end of part one of this tutorial, I promised we would learn how to define new primitives to do special tasks. To better understand what is meant by this, imagine a geometry student who has been asked to find the difference between the area and the circumference of five circles with radii of 1 through 5. To do this, she must type:

```
Input: (setq pi 3.14)
Value: 3.14
Input: (difference (times pi 1 1) (times 2 pi 1))
Value: -3.14
Input: (difference (times pi 2 2) (times 2 pi 2))
Value: 0
Input: (difference (times pi 3 3) (times 2 pi 3))
Value: 9.42
```

...

And so forth. It is obviously going to be a very tedious task. It would be much easier for this student to do her work if Lisp had built in functions to calculate area and circumference. However, since Lisp does not have these functions, we can write them ourselves using `de`:

```
Input: (de area (r)
(times pi r r))
Value: area
Input: (de circumference (r)
(times 2 pi r))
Value: circumference
```

Let's look closely at the first definition. The first argument to `de` is the name of a symbol. This is the symbol that will represent our new function, just as the first argument to `setq` is the symbol to represent our

value. The second argument is a list of symbols. Although there is only one symbol in the list, other function definitions will have many more. The symbols in this list represent the arguments to our new function. The function area will take only one argument which we have labeled `r`.

The remaining argument to `de` is the actual definition of our function. It says to take `r`, the argument given in an expression that calls the area function, and multiply it by itself and then by `pi`. We can use `pi` to represent 3.14 because Lisp remembers the values of symbols even inside function definitions.

The definition for circumference is similar, except that it says to multiply its argument by `pi` and then by 2. With both of these new functions, the student need only type (difference (area 1) (circumference 1)) to get the answer to her first problem. Using these user-defined primitives instead of the standard Lisp functions not only reduces the amount she has to type but it makes her code much more readable. We can tell exactly what the expression is doing just by glancing at it.

Of course, user-defined functions do not have to operate on numbers. Using the list processing functions `car` and `cdr` that we learned last month together with `de`, we can write functions that return the second, third, or fourth element of a list:

```
Input: (de second (list)
(cadr list))
Value: second
Input: (de third (list)
(caddr list))
Value: third
Input: (de fourth (list)
(cadddr list))
```

Remember that (cadr list) is the same as (car (cdr list)), (caddr list) is the same as (car (cdr (cdr list))), and (cadddr list) is the same as (car (cdr (cdr (cdr list)))).

Before closing this second part of the tutorial, there are a few more list processing functions that we should cover. We will be concentrating on lists for a few months before returning to numbers.

```
Input: (setq a '(1 2 3))
Value: (1 2 3)
Input: (reverse a)
Value: (3 2 1)
Input: (setq b '(4 5 6))
Value: (4 5 6)
Input: (append a b)
Value: (1 2 3 4 5 6)
Input: (setq c '(3 4 5))
```


Value: (3 4 5)
Input: (append a c)
Value: (1 2 3 3 4 5)
Input: (union a c)
Value: (1 2 3 4 5)

Note the subtle difference between append and union. **Append** blindly concatenates lists while **union** makes sure only elements of the second list, which are not already in the first list, are added.

Here are some functions that work a little differently than those we have already discussed:

Input: (equal a b)
Value: nil
Input: (equal a '(1 2 3))
Value: t
Input: (equal a a)
Value: t
Input: (member 1 a)
Value: (1 2 3)
Input: (member 2 a)
Value: (2 3)
Input: (member 4 a)
Value: nil

The equal function is fairly straight forward. If the two arguments are the same it returns t and if they are different it returns nil. T and nil are Lisp's generic values for "true" and "false".

The member function is a little more complicated. If the first argument is contained in the list given as the second argument, member returns the portion of the list that begins with the first argument. If the first argument is not an element of the list, member returns nil.

We call t the generic truth value because Lisp uses many things to represent "true". In fact, any value other than nil can be used to mean truth. The member function could simply return t if the first argument was in the list, but the writers of Lisp thought it might be useful to know not only whether the first argument was in the list but where it actually was.

We've covered a lot this month. Still, we have not written anything that looks much like a "program". We will begin to do that next month when we look at ways of making our user-defined functions more powerful. This will include conditional expressions and various methods of looping. Some of the concepts you will recognize from languages like BASIC, Pascal, and C, but others are unique to Lisp. You will quickly realize, however, that all of these primitives that appear to be control structures are really just more complicated list processing functions. This is because, in Lisp, even our programs are just big lists.

•AC•

AMIGA \$9.95 ea.

Public Domain Disks

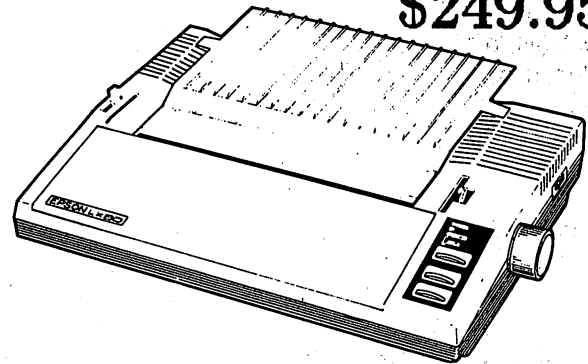
Disk 1 Star Trek (Text), EZ-Terminal,
Amiga Doodle

Disk 2 Copperstate Copier for 2 drives

Disk 3 Amiga Info

EPSON® LX-80 Printer

\$249.95



JX-80 Colored Printer Ribbons

\$14.95

Amiga Printer Cables \$29.95

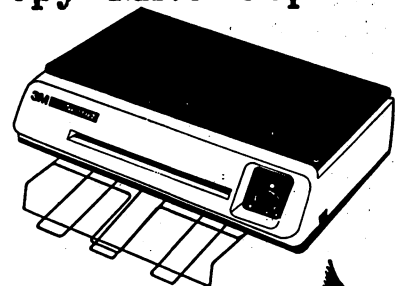
3M Copy Mite Copiers

Copy Mite 1

\$169.95

Copy Mite 2

\$269.95



Cardinal Software

13646 Jeff Davis Hwy.

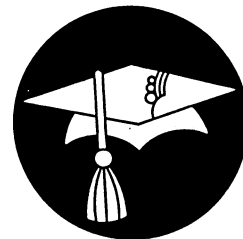
Woodbridge, VA 22191



Order NOW! 800 762-5645

Info (703) 491-6502 SHIPPING EXTRA





Commodore's Great Educational Advantage

On January 29th, Commodore announced a 25% discount on Amiga computer systems to every university, faculty member, and college student in the United States.

In a move that resembled Apple Computer's Educational Consortium, Commodore has tried to reach the huge market of college students. This is seen as a good sign for all Amiga owners (even those that cannot buy through the college program) for four reasons:

1. Commodore is placing an untold number of machines into the hands of individuals who have traditionally produced the most enthusiasm and resulting products for computers, the American College Student and University.
2. Commodore has sparked interest in the Amiga in the educational sector. Faculty and students, who had previously never considered buying or utilizing an Amiga, are now giving the machine a very thoughtful look.
3. Amiga's base of users should grow dramatically, producing stronger Amiga markets in software and hardware. This should encourage manufacturers to produce and sell Amiga products. Companies that have been waiting to see "how it does", will now have a broader consumer base for their products.
4. Commodore will establish Amiga's position in the computer and educational communities in much the same manner Apple Computer, Inc. established the Macintosh™.

However, Commodore has again taken notes from Apple Computer's past mistakes and has rewritten the rules for their college promotion.

Where Apple Computer selected only a handful of schools throughout the country as participants in their program, Commodore has shipped over 40,000 kits announcing the plan to colleges and universities in the US. From the information we have been able to obtain on the program, every college student and faculty member is eligible for the discount. They must obtain a Commodore certificate from their university or college and give it to their local Amiga dealer.

Again, unlike Apple's plan of dumping large quantities of Macintosh Computers into selected areas and bypassing the local dealers, Commodore has placed the discount program in the hands of the dealers. The dealer will give the participant the discount and then be reimbursed by products from Commodore. In this way, the Amiga owner becomes accustomed to dealing with their local dealer. This should help the customer by better support, and the dealer by increased sales and traffic.

The announced plan was on the Amiga 1000 personal computer which includes the internal disk drive, the Amiga 1080 Color Monitor, and the additional 256K ram Expansion Cartridge. There have been some hints that this discount may extend into all or most of the products that Commodore is now marketing for the Amiga.

The program deadline is June 30 and Amiga dealers are reporting a great deal of interest from local universities. At press time, the final details of this plan were being routed to dealers. Our initial reaction from dealers has been very enthusiastic.

From the information available, it is apparent that Commodore wants to place Amigas where they will do the most good for the entire Amiga user community.

•AC•

Attention:

Amiga User Groups

If you belong to an Amiga User Group, Amazing Computing™ wants to know.

Amazing Computing is interested in the Amiga user and Amiga User Groups are one of the best investments of time and energy that an Amiga owner can make. The opportunity to share challenges and discoveries with other Amiga owners can help you solve some difficult problems and generate a few great applications for your Amiga.

To help in the growth of a network of Amiga user groups, Amazing Computing™ will publish a list of groups in our future issues. Any Amiga owner in your area, will have an opportunity to contact your group and join. There is strength in numbers and a good club will always help.

If you have a club and would like Amazing Computing to list it, please send the club's name, mailing address, meeting place and time, with a contact's telephone number to:

**Amiga User Groups
Amazing Computing
PiM Publications
P.O. Box 869
Fall River, MA 02722**

If you do not belong to an Amiga user group, watch for our listing.....

Or better yet,

Create your own Amiga user group!

Amiga

Software

Languages & Tools

| | | | |
|-----------------------------------|-----------------------------|----------|-----------|
| Turbo Pascal™ | Borland International | TBA | TBA |
| Amiga TLC Logo | Commodore Business Machines | \$99.95 | NOW |
| Amiga Pascal | Commodore Business Machines | \$99.95 | NOW |
| Amiga Assembler | Commodore Business Machines | \$99.95 | NOW |
| Amiga Lisp | Commodore Business Machines | \$199.95 | NOW |
| ABasic | Commodore Business Machines | \$24.95 | NOW |
| WACK S/W Toolkit | Commodore Business Machines | \$99.95 | NOW |
| IBM PC Cross Development Package | Commodore Business Machines | \$99.95 | NOW |
| MultiForth™ | Creative Solutions Inc. | \$179.00 | 1 Qtr '86 |
| Marauder (Disk utility)™ | Discovery Software | \$39.95 | NOW |
| Exactly! | Discovery Software | \$24.95 | Mar '86 |
| The Explorer | Interactive Analytic Node | \$35.00 | NOW |
| Amiga C Compiler™ | Lattice | \$149.95 | NOW |
| Lattice MacLibrary™ | Lattice | \$100.00 | NOW |
| Lattice Make Utility™ | Lattice | \$125.00 | NOW |
| Text Utilities™ | Lattice | \$149.95 | NOW |
| Lattice Screen Editor™ | Lattice | \$100.00 | NOW |
| Panel (screen Layout utilities)™ | Lattice | \$195.00 | FEB '86 |
| Cross Compiler (MS-Dos to Amiga)™ | Lattice | \$250.00 | NOW |
| dBCIII (Library of DB functions)™ | Lattice | \$150.00 | NOW |
| Cross Reference Generator™ | Lattice | \$45.00 | NOW |
| Aztec C Developer Edition | MANX | \$299.00 | Mar '86 |
| Aztec C Commercial Edition | MANX | \$499.00 | Mar '86 |
| MCC Pascal™ | Metacomco | \$99.95 | NOW |
| Cambridge Lisp™ | Metacomco | \$199.95 | NOW |
| MetaScope | Metadigm, Inc. | \$95.00 | NOW |
| MetaScribe | Metadigm, Inc. | \$85.00 | NOW |
| MetaTools | Metadigm, Inc. | \$69.95 | NOW |
| Programmer's Editor | Micro Forge | \$69.95 | Mar '86 |
| Ram Disk | Micro Forge | \$24.95 | Mar '86 |
| TxEd | Microsmith | \$39.95 | NOW |
| Custom Printer Driver Text only | Software Supermarket | \$25.00 | NOW |
| Graphics upgrade | Software Supermarket | \$5.00 | NOW |
| UBZ Forth™ | UBZ Software | \$85.00 | Jan '86 |
| ADFO"Disk File Organizer" | Westcom Ind. | \$59.95 | Apr '86 |
| Amermtest | Westcom Ind. | \$49.95 | 2 Qtr '86 |
| ZLI | Zoxso | \$69.95 | NOW |

Products

Entertainment

| | | | |
|---|-------------------------------------|---------|----------|
| Mindshadow™ | Activision | \$44.95 | NOW |
| Borrowed Time™ | Activision | \$44.95 | NOW |
| Hacker | Activision | \$44.95 | NOW |
| Musicraft | Commodore Business Machines | TBA | Apr '86 |
| Robot War | Commodore Business Machines | \$49.95 | TBA |
| Wynd Walker | Commodore Business Machines | \$49.95 | Mar '86 |
| Driving Game | Commodore Business Machines | \$49.95 | TBA |
| Julius Erving, Larry Bird, One on One™ | Electronic Arts, Inc. | \$35.95 | NOW |
| The Seven Cities of Gold™ | Electronic Arts, Inc. | \$39.95 | NOW |
| Arcon™ | Electronic Arts, Inc. | \$39.95 | NOW |
| Skyfox™ | Electronic Arts, Inc. | \$39.95 | Mar '86 |
| Adventure Construction Set™ | Electronic Arts, Inc. | \$49.95 | Mar '86 |
| Articfox™ | Electronic Arts, Inc. | \$39.95 | Mar '86 |
| Software Golden Oldies (ILife, Pong, Eliza, Adventure) | Electronic Arts, Inc. | \$34.95 | Mar '86 |
| Sargon III™ | Hayden Software | \$49.95 | TBA |
| Zork I™ | Infocom, Inc. | \$39.95 | Now |
| Zork II™ | Infocom, Inc. | \$49.95 | TBA |
| Wishbringer™ | Infocom, Inc. | \$39.95 | NOW |
| Enchanter™ | Infocom, Inc. | \$39.95 | NOW |
| Deadline™ | Infocom, Inc. | \$49.95 | NOW |
| Witness™ | Infocom, Inc. | \$39.95 | NOW |
| Suspect™ | Infocom, Inc. | \$39.95 | NOW |
| Starcross™ | Infocom, Inc. | \$49.95 | NOW |
| Suspended™ | Infocom, Inc. | \$49.95 | NOW |
| Planetfall™ | Infocom, Inc. | \$39.95 | NOW |
| A Mind Forever Voyaging™ | Infocom, Inc. | \$44.95 | NOW |
| Infidel™ | Infocom, Inc. | \$44.95 | NOW |
| Seastalker™ | Infocom, Inc. | \$39.95 | NOW |
| Cut Throat | Infocom, Inc. | \$39.95 | NOW |
| Hitchhiker's Guide to the Galaxy™ | Infocom, Inc. | \$39.95 | NOW |
| Spellbreaker™ | Infocom, Inc. | \$49.95 | NOW |
| Sorcerer™ | Infocom, Inc. | \$39.95 | NOW |
| Keyboard Cadet™ | Mindscape | \$39.95 | Mar '86 |
| Racter™ | Mindscape | \$44.95 | NOW |
| Halley Project™ | Mindscape | \$44.95 | Mar '86 |
| Deja Vu™ | Mindscape | \$49.95 | June '86 |
| Barataccas™ | Mindscape | \$49.95 | NOW |
| Jet | Sublogic Communications Corporation | TBA | Mar '86 |

MORE AMIGA

Business

Rags to Riches™

Ledger

Receivables

Payables

3 Pac

Amiga Calcraft

Textcraft

Transformer

Financial Cookbook™

Unicalc™

Maxiplan

Analyze

Scribble!

VIP

Chang Labs

Chang Labs

Chang Labs

Chang Labs

Chang Labs

Commodore Business Machines

Commodore Business Machines

Commodore Business Machines

Electronic Arts, Inc.

Lattice

Maxicorp

Micro-Systems Software, Inc.

Micro-Systems Software, Inc.

VIP Technologies

\$199.95

\$199.95

\$199.95

\$499.50

\$99.95

\$99.95

\$99.95

\$79.95

\$79.95

\$125.00

\$99.95

\$69.95

\$199.95

NOW

NOW

NOW

NOW

Mar '86

NOW

Apr '86

NOW

NOW

Feb '86

Jan '86

NOW

Mar '86

Communications

AmigaTerm

Telecraft

EITerm™

Maxicom™

OnLine

BBS-PC™

Commodore Business Machines

Commodore Business Machines

Elcom Software

Electronic Arts/Maxicorp

Micro-Systems Software, Inc.

Micro-Systems Software, Inc.

\$49.95

\$79.95

\$75.00

\$49.95

\$69.95

\$99.95

TBA

TBA

1Qtr.'86

NOW

NOW

JAN '86

Graphics

Aegis Animator™ with Aegis Images™

Aegis Images™

Amiga Draw™

Amiga Impact™

Amiga Pro Draw

Graphicraft

DeluxePaint™

Sound Vision™

Aegis

Aegis

Aegis

Aegis

Aegis

Commodore Business Machines

Electronic Arts, Inc.

Hayden Software

\$139.95

\$79.95

\$199.00

TBA

TBA

\$49.95

\$79.95

TBA

Jan '86

Jan '86

Feb '86

TBA

TBA

NOW

NOW

TBA

Products!

Hardware

| | | | |
|--|------------------------------|-----------|-------------|
| A-Time™ | Akron Systems Development | \$49.95 | NOW |
| 2Meg-8Meg Card | Akron Systems Development | TBA | TBA |
| External 5.25 Disk Drive | Commodore Business Machines | \$395.00 | TBA |
| External 3.5 Disk Drive | Commodore Business Machines | \$295.00 | NOW |
| Amiga Transformer | Commodore Business Machines | TBa | Apr '86 |
| Modem/1200 RS | Commodore Business Machines | \$295.00 | TBA |
| 256K Ram Expansion | Commodore Business Machines | \$195.00 | NOW |
| RF Modulator | Commodore Business Machines | \$49.95 | NOW |
| Amiga Answer Mate | Commodore Business Machines | \$249.95 | Apr '86 |
| Answer Mate Modem Cartridge | Commodore Business Machines | \$99.95 | Apr '86 |
| Genlock Interface | Commodore Business Machines | \$249.95 | Apr '86 |
| Midi Interface | Commodore Business Machines | \$49.95 | NOW |
| Amiga Live! | Commodore Business Machines | TBA | Apr '86 |
| 68020 Processor Board | Computer System Associates | N/A | N/A |
| 68020 Processor Board | IO Inc. | N/A | N/A |
| Penmouse+™ | Kurta | \$375 | NOW |
| Series ONE™ 200 PPI | Kurta | | |
| 8.5 x 11 | Kurta | \$695 | NOW |
| 12 x 12 | Kurta | \$795 | NOW |
| 12 x 17 | Kurta | \$895 | NOW |
| Series TWO™ 12 x 12 | Kurta | | |
| 400 PPI | Kurta | \$835 | NOW |
| 1000 PPI | Kurta | \$895 | NOW |
| 40-Meg Hard Disk (use with expansion box) | Micro Forge | \$1479.95 | NOW |
| 2nd 10-Meg Hard Disk | Micro Forge | \$755.95 | NOW |
| 2nd 20-Meg Hard Disk | Micro Forge | \$849.95 | NOW |
| 2nd 40-Meg Hard Disk | Micro Forge | \$1099.95 | NOW |
| Stereo Sound Digitizer | Micro Forge | \$344.95 | Mar '86 |
| 2 Megabyte Memory Board | Micro Forge | \$859.95 | Mar '86 |
| 2400B Modem | Micro-Systems Software, Inc. | \$429.00 | NOW |
| T-Connect™ | Tecmar | \$450.00 | NOW |
| T-Card (256K)™ | Tecmar | \$795.00 | NOW |
| T-Card (512K)™ | Tecmar | \$895.00 | NOW |
| T-Card (1Meg)™ | Tecmar | \$995.00 | NOW |
| T-disk (20Meg)™ | Tecmar | \$1495.00 | NOW |
| T-modem™ | Tecmar | \$695.00 | NOW |
| T-tape™ | Tecmar | TBA | 2nd Qtr '86 |

AMIGA Books

The Amiga Technical Reference Series

| | | | |
|--|-------------------------------|---------|----------|
| The Amiga Hardware Reference Manual | Addison-Wesley Publishing Co. | \$24.95 | Mar '86 |
| The Amiga Rom Kernel Manual: | | | |
| Library and Devices | Addison-Wesley Publishing Co. | \$29.95 | Mar '86 |
| Exec | Addison-Wesley Publishing Co. | \$19.95 | Apr '86 |
| The Amiga Intuition Reference Manual | Addison-Wesley Publishing Co. | \$24.95 | Mar '86 |
| The AmigaDos™ User's Manual | Bantam Books | \$24.95 | Mar '86 |
| AmigaDos User's Manual | Commodore Business Machines | \$14.95 | NOW |
| Amiga Technical Documentation | Commodore Business Machines | \$99.95 | NOW |
| Contains a set of six technical manuals: | | | |
| AmigaDos Developer's Manual | | | |
| AmigaDos Technical Reference Manual | | | |
| AmigaDos User's Manual | | | |
| Amiga Hardware Manual | | | |
| Amiga Rom Kernel Manual | | | |
| Intuition: The Amiga User Interface | | | |
| The Official Amiga Programmer's | | | |
| Reference Guide | Sybx Computer Books | \$28.95 | June '86 |
| The Amiga Programmer's Handbook | Sybx Computer Books | \$24.95 | Mar '86 |

Don't Be Shy!
Communicate from your

AMIGA

With ElTerm,

you can open your Amiga up to the world of telecommunications. ElTerm is a complete package for terminal emulation and file transfer. ElTerm can upload and download files using Kermit and Xmodem protocols.

ElTerm will be available first quarter 1986.
Suggested retail price is \$75.00 US
Dealer Inquiries Invited.

Elcom Software

16 Oak St. #2
New Brunswick, NJ 08901

Amiga is a trademark of Commodore-Amiga Inc.
ElTerm is a trademark of Elcom Software.

**Haven't You Set Your AMIGA'S
Time And Date Once Too Often?**

Introducing

A - T I M E

*A clock/calendar card with battery back-up,
so you will never have to set the time and date
in your AMIGA, EVER AGAIN!*

- Plugs into the parallel port.
- A completely transparent printer port is provided, with total compatibility to all I/O operations.
- Battery back-up keeps the clock/calendar date valid on power down.
- Custom case with a footprint of only 2 1/4" x 7/8" x 3 1/4" (W x D x H) in standard AMIGA color.
- Leap year capability.
- A - T I M E package contains:
1-A - T I M E clock/calendar module
1-3.5" DS Utilities Disk
Operating instructions

PRICE \$49⁹⁵

AVAILABLE: NOW

Mail check to:

AKRON SYSTEMS DEVELOPMENT (ASD)
P. O. BOX 6408 (409) 833-2686
BEAUMONT, TEXAS 77705

include \$3.50 for shipping and handling
For MC/VISA orders call (409) 833-2686
AMIGA is a trademark of Commodore - Amiga inc.

Attention:

Amiga

Developers

If you are an Amiga developer with an Amazing Amiga product, Amazing Computing can help. To assist in the development of other startup companies, PiM Publications and Amazing Computing have offered advertising rates to Amiga developers at ridiculously low prices.

Our aim is to create a forum for developers in Amazing Computing. We believe we can help the Amiga market produce good, productive software if new development companies have an opportunity to reach their customers without using all of their resources. Therefore, we have offered advertising rates far below what other publications can offer. And our magazine is seen and read by the people Amiga developers want to reach, Amiga owners and Amiga dealers.

Our ridiculously low ad rates are only in effect for a limited time. If you are developing a product for this fast growing market, please take advantage of these rates while we can still offer them. For more information contact:

**Advertising Sales
Amazing Computing
PiM Publications
P.O. Box 869
Fall River, MA 02722
617-679-3109**

Amazing Computing, your Amiga resource.



SuperTerm V1.0

by Kelly Kauffman

CIS 70206,640

Welcome to Amiga Basic from Commodore and Microsoft and to the wonderful world of communications with your Amiga. The following program will allow you to effectively communicate between your Amiga and other systems with ease.

This program has many more features than did its predecessor, EZ-TERM. However, the functions are VERY straight-forward and should be easy to understand and use. Again, this program is not intended for the person who just took their Amiga out of the box yesterday---A good working knowledge of AmigaDos is not required, but suggested.

Unfortunately, the Xmodem Send and Receive is again bugged. Not in this program, but in the driver program supplied on your Workbench disk that "runs" the Serial port. It seems that ALL CTRL-S's and CTRL-Q's are automatically removed and never make it through the serial port. It stops CTRL-S's from being sent, and will not receive CTRL-S or CTRL-Q. This creates a problem because files contain CTRL-S's and Q's and must be passed through to Microsoft Basic in order to properly complete the Xmodem Checksums and receive the file correctly. For this reason the Xmodem Send and Receive does not function properly.

However, Xmodem has been left in SuperTerm with hope Commodore-Amiga will fix this bug in Workbench Revision 1.2. No modifications will be necessary at that time in order to use the Xmodem areas of the program since they are just waiting for the correctly written driver. After you receive WB 1.2, simply load SuperTerm and go!

One disadvantage to this program is the relatively small buffer size, making large file transfers impossible. I hope to eventually get around this (though efforts so far leave me with "Out of Memory" errors) I will keep you posted and will hopefully publish a fix to this soon.

Also, In the macro section, if you end your macro in [RET], a Carriage Return will be sent after that macro key is pressed (Function Keys 1-10). If the [RET] is omitted, no Carriage Return will be sent, and you will have to manually depress the RETURN key.

For a detailed description as to the terminology used in the program, refer to AMAZING COMPUTING™, Premiere Issue, Page 11 for details.

•AC•



SUPERTERM in AMIGA BASIC

by Kelly Kauffman
CIS 70206,640

```
REM - Superterm V1.0
REM - by Kelly Kauffman
REM - January 3, 1985
REM
```

```
PRINT "Do you want CTRL-C's to end the program
      (E) or actually send a CTRL-C"
PRINT "through the modem (S) Input Choice
      (Either E or S)";
INPUT choice$
IF choice$="S" OR choice$="s" THEN
  BREAK ON
  ON BREAK GOSUB breaker
END IF
DIM mac$(10)
DIM bf$(256)
GOSUB getterm
WIDTH 77
rcp(1)=1
fl(1)=1
fl(2)=1
cp(2)=1
DIM bd(12)
F$="COM1:"
```

```
menusel=3
GOSUB baud
menusel=1
GOSUB trm
menusel=3
GOSUB trm
menusel=6
GOSUB trm
menusel=4
GOSUB decipher
menusel=9 : 'databits
GOSUB decipher
menusel=11 : 'stopbits
GOSUB decipher
GOSUB initterm
GOSUB initmenu
MENU ON
ON MENU GOSUB menustuff
```

term:

```
ON ERROR GOTO oops
PRINT
" ----->>>> Terminal Mode <<<<-----"
MENU ON
```

20

```

IF LOC(2)=1 THEN
  IF LOC(2)=1 THEN
in$=INPUT$(1,2)
  IF ASC(in$)=13 AND lf=1 THEN
PRINT
  PRINT in$;
  IF cap=1 THEN
    buff$=buff$+in$
    IF lf=1 THEN buff$
      =buff$+CHR$(10)
  END IF
END IF
b$=INKEY$
IF b$<>" " THEN
  IF ASC(b$)>128 AND
    ASC(b$)<139 THEN
    GOTO macro
  PRINT #2,b$;
  IF dp=1 THEN PRINT b$;
END IF
GOTO 20

```

cap:

```

PRINT "Capture ";
IF cap=1 THEN
  PRINT "ON"
  PALETTE 0,.78,.05,.05
ELSE
  PRINT "OFF"
  PALETTE 0,.05,.05,.78
END IF
RETURN

```

linfd:

```

PRINT "Linefeeds";
IF lf=1 THEN PRINT "ON" ELSE
PRINT "OFF"
RETURN

```

sav:

```

PRINT
PRINT
PRINT
  "Save Buffer to what filename ";
INPUT file$
CLOSE 5
OPEN file$ FOR OUTPUT AS 5
PRINT #5,buff$;
CLOSE 5
RETURN

```

lod:

```

PRINT "Load Buffer"
PRINT
PRINT
PRINT "What filename to load";
INPUT file$
CLOSE 5
OPEN file$ FOR INPUT AS #5
PRINT
PRINT "Length of File = ";LOF(5)
buff$=INPUT$(LOF(5),5)
PRINT
PRINT "Buffer Load Completed."
PRINT
RETURN

```

initmenu:

```

MENU 1,0,1,"Misc. "
MENU 1,1,1,"Save Buffer "
MENU 1,2,1,"Load Buffer "
MENU 1,3,1,"Edit Macros "
MENU 1,4,1,"Review Buffer "
MENU 1,5,1,"Show Info "
MENU 1,6,1,"Clear Buffer "
MENU 1,7,1,"Disk Directory"
MENU 1,8,1,"Default Direct"
MENU 1,9,1,"Delete File "
MENU 1,10,1,"Rename File "
MENU 1,11,1,"Print Free Mem"
MENU 2,0,1,"Term Options"
MENU 2,1,cp(1)," Capture Off"
MENU 2,2,cp(2)," Capture On"
MENU 2,3,fl(1)," LF Sim. Off"
MENU 2,4,fl(2)," LF Sim. On"
MENU 2,5,dp(1)," Half Duplex"
MENU 2,6,dp(2)," Full Duplex"
MENU 3,0,1,"Transmit"
MENU 3,1,1,"Normal Upload "
MENU 3,2,1,"Prompt Upload "
MENU 3,3,1,"Xmodem Receive"
MENU 3,4,1,"Xmodem Send "
MENU 4,0,1,"Baud Rate"
MENU 4,1,bd(1)," 110 "
MENU 4,2,bd(2)," 150 "
MENU 4,3,bd(3)," 300 "
MENU 4,4,bd(4)," 600 "
MENU 4,5,bd(5)," 1200 "
MENU 4,6,bd(6)," 1800 "
MENU 4,7,bd(7)," 2400 "
MENU 4,8,bd(8)," 3600 "
MENU 4,9,bd(9)," 4800 "
MENU 4,10,bd(10)," 7200 "
MENU 4,11,bd(11)," 9600 "

```

```

MENU 4,12,bd(12)," 19200 "
MENU 5,0,1,"Config"
MENU 5,1,0,"Parity"
MENU 5,2,par(1)," Odd "
MENU 5,3,par(2)," Even "
MENU 5,4,par(3)," None "
MENU 5,5,0,"Data-Bits"
MENU 5,6,db(1)," 5 "
MENU 5,7,db(2)," 6 "
MENU 5,8,db(3)," \ "
MENU 5,9,db(4)," 8 "
MENU 5,10,0,"Stop-Bits"
MENU 5,11,sb(1)," 1 "
MENU 5,12,sb(2)," 2 "
RETURN

```

file:

```

ON menusel GOSUB
sav,lod,ed,review,info,clearbuf,direct
,chdf,killer,ren,mem
GOTO term

```

trm:

```

IF menusel=1 THEN
cp(1)=2:cp(2)=1
  IF menusel=2 THEN
cp(2)=2:cp(1)=1
    IF menusel=3 THEN fl(1)=2:fl(2)=1
    IF menusel=4 THEN fl(2)=2:fl(1)=1
    IF menusel=5 THEN
dp(1)=2:dp(2)=1
      IF menusel=6 THEN
dp(2)=2:dp(1)=1
        IF first=1 THEN
          MENU 2,0,1,"Term Options"
          MENU 2,1,cp(1)," Capture Off"
          MENU 2,2,cp(2)," Capture On"
          MENU 2,3,fl(1)," LF Sim. Off"
          MENU 2,4,fl(2)," LF Sim. On"
          MENU 2,5,dp(1)," Half Duplex"
          MENU 2,6,dp(2)," Full Duplex"
        END IF
        IF first=1 THEN
          IF menusel=1 THEN cap=0:GOTO
cap
            IF menusel=2 THEN cap=1:GOTO
cap
              IF menusel=3 THEN lf=0:GOTO
linfd
                IF menusel=4 THEN lf=1:GOTO
linfd
                  IF menusel=5 THEN dp=1:GOTO
duplex
                    IF menusel=6 THEN dp=0:GOTO
duplex

```

| | | |
|--|---|--|
| <pre> END IF RETURN baud: GOSUB clearbaud ON menu\$ GOSUB bd110,bd150,bd300,bd600, bd1200,bd1800,bd2400, bd3600,bd4800,bd7200, bd9600,bd19200 IF first=1 THEN MENU 4,0,1,"Baud Rate" MENU 4,1,bd(1)," 110 " MENU 4,2,bd(2)," 150 " MENU 4,3,bd(3)," 300 " MENU 4,4,bd(4)," 600 " MENU 4,5,bd(5)," 1200 " MENU 4,6,bd(6)," 1800 " MENU 4,7,bd(7)," 2400 " MENU 4,8,bd(8)," 3600 " MENU 4,9,bd(9)," 4800 " MENU 4,10,bd(10)," 7200 " MENU 4,11,bd(11)," 9600 " MENU 4,12,bd(12)," 19200 " END IF IF first=1 THEN GOSUB initterm ELSE RETURN RETURN bd110: baud\$="110" bd(1)=2 RETURN bd150: baud\$="150" bd(2)=2 RETURN bd300: baud\$="300" bd(3)=2 RETURN bd600: baud\$="600" bd(4)=2 RETURN bd1200: baud\$="1200" </pre> | <pre> bd(5)=2 RETURN bd1800: baud\$="1800" bd(6)=2 RETURN bd2400: baud\$="2400" bd(7)=2 RETURN bd3600: baud\$="3600" bd(8)=2 RETURN bd4800: baud\$="4800" bd(9)=2 RETURN bd7200: baud\$="7200" bd(10)=2 RETURN bd9600: baud\$="9600" bd(11)=2 RETURN bd19200: baud\$="19200" bd(12)=2 RETURN clearbaud: FOR i=1 TO 12 bd(i)=1 NEXT i RETURN initterm: com\$=F\$ comma\$="," </pre> | <pre> com\$=com\$+baud\$ com\$=com\$+comma\$ com\$=com\$+parity\$ com\$=com\$+comma\$ com\$=com\$+databit\$ com\$=com\$+comma\$ com\$=com\$+stopbits\$ first=1 ON ERROR GOTO oops CLOSE 2 OPEN com\$ AS #2 LEN=128 CLS PRINT "RS232 configured to: "; PRINT baud\$;" Baud - "; IF parity\$="N" THEN PRINT "No Parity"; IF parity\$="E" THEN PRINT "Even Parity"; IF parity\$="O" THEN PRINT "Odd Parity"; PRINT "- "; PRINT " ";databit\$;" Databits - "; PRINT stopbits\$;" Stopbit(s)."; PRINT RETURN parity: par(1)=1 par(2)=1 par(3)=1 par(menu\$)=2 IF menu\$=1 THEN parity\$="O" IF menu\$=2 THEN parity\$="E" IF menu\$=3 THEN parity\$="N" IF first=1 THEN GOSUB initconfig IF first=1 THEN GOSUB initterm RETURN databits: db(1)=1 db(2)=1 db(3)=1 db(4)=1 db(menu\$)=2 IF menu\$=1 THEN databit\$="5" IF menu\$=2 THEN databit\$="6" IF menu\$=3 THEN databit\$="7" IF menu\$=4 THEN databit\$="8" IF first=1 THEN GOSUB initconfig IF first=1 THEN GOSUB initterm RETURN stopbits: sb(1)=1 sb(2)=1 </pre> |
|--|---|--|

```

sb(menuusel)=2
IF menuusel=1 THEN stopbits$="1"
IF menuusel=2 THEN stopbits$="2"
IF first=1 THEN GOSUB initconfig:
  GOSUB initerm
RETURN

```

menustuff:

```

menuid=MENU(0)
menuusel=MENU(1)
ON menuid GOSUB file,term,trans,
  baud,decipher
MENU ON
RETURN

```

oops:

```

CLOSE #5
BEEP
IF ERR=64 THEN PRINT
  " Invalid RS232 Model!"
IF ERR=64 THEN PRINT
  " Try Other Paramaters"
IF ERR<>64 THEN PRINT
  "Error - ";ERR:MENU ON:
  RESUME term
PRINT
PRINT
MENU ON
10 IF MENU(0)<>0
  THEN RESUME erormenu
  ELSE GOTO 10

```

erormenu:

```

menuusel=MENU(1)
ON menuid GOSUB file,term,trans,
  baud,decipher
MENU ON
GOTO term

```

decipher:

```

IF menuusel<5
  THEN menuusel=menuusel-1:
  GOTO parity
IF menuusel<10
  THEN menuusel=menuusel-5:
  GOTO databits
IF menuusel<13
  THEN menuusel=menuusel-10:
  GOTO stopbits

```

initconfig:

```

MENU 5,0,1,"Config"

```

```

MENU 5,1,0,"Parity"
MENU 5,2,par(1)," Odd "
MENU 5,3,par(2)," Even "
MENU 5,4,par(3)," None "
MENU 5,5,0,"Data-Bits"
MENU 5,6,db(1)," 5 "
MENU 5,7,db(2)," 6 "
MENU 5,8,db(3)," 7 "
MENU 5,9,db(4)," 8 "
MENU 5,10,0,"Stop-Bits"
MENU 5,11,sb(1)," 1 "
MENU 5,12,sb(2)," 2 "
RETURN

```

trans:

```

IF menuusel=1 THEN normup
IF menuusel=2 THEN prmptup
IF menuusel=3 THEN xrec
IF menuusel=4 THEN xsend

```

normup:

```

In=LEN(buff$)
FOR i=1 TO In
  PRINT #2,MID$(buff$,i,1);
  IF INKEY$<>"" THEN
    PRINT
    PRINT
    PRINT " Aborted!!!!!"
    PRINT
    GOTO term
  END IF
NEXT i
PRINT
PRINT " Normal Buffer Upload
  Completed."
PRINT
GOTO term

```

prmptup:

```

In=LEN(buff$)
FOR i=1 TO In
  PRINT #2,MID$(buff$,i,1);
  IF ASC(MID$(buff$,i,1))=13
    THEN GOSUB waitprompt
  IF INKEY$<>"" THEN
    PRINT
    PRINT
    PRINT " Aborted !!!"
    PRINT
    GOTO term
  END IF
NEXT i
GOTO term

```

waitprompt:

```

IF LOC(2)=1 THEN qw$=
INPUT$(1,2)
  IF qw$<>prompt$
    THEN waitprompt ELSE
    RETURN
END IF

```

xrec:

```

PRINT
PRINT "Receive what
Filename";:INPUT file$
IF file$="" THEN abort
CLOSE #3
OPEN file$ FOR OUTPUT AS #3
timeout=500
ack$=CHR$(6)
nak$=CHR$(21)
eot$=CHR$(4)
blawck=1
n=1
eotflag=0
jr$=nak$:GOSUB princnr
30 bf$="" :t=0:r$="":
  tim=0:chsum=0
  FOR t=1 TO 132
50  r$=""
    IF LOC(2)=1 THEN
      r$=INPUT$(1,2)
      bf$=bf$+r$
      tim=0
      PRINT ". ";
    ELSE
      tim=tim+1
    END IF
    IF t=1 THEN IF bf$=eot$
      THEN 40
    IF tim>timeout THEN
      PRINT "Timeout"
      jr$=nak$
      GOSUB princnr
      t=0
      tim=0
      GOTO 30
    END IF
    IF r$="" THEN 50
  NEXT t
  chsum=0
  GOSUB checksummer
  IF chsum=ASC(RIGHT$(bf$,1))
    THEN
      PRINT #3,MID$(bf$,4,127);
      PRINT " Received Block ";
      blawck jr$=ack$:
      GOSUB princnr
      PRINT
      blawck=blawck+1

```



```

GOTO 30
ELSE
  BEEP
  PRINT "Checksum Error in
    Block ";blawck
  PRINT "I get ";chsum;
  PRINT ". And He get's ";
    ASC(RIGHT$(bf$,1))
  jd$=ack$:GOSUB princchr
  PRINT
END IF
GOTO 30

40 CLOSE #3

PRINT #2,ack$;
PRINT "File Transfer is Completed."
GOTO term

checksum:

chsum=0
FOR i=1 TO 128
  chsum=chsum+ASC(MID$(
    (bf$,i,1)) AND 255
NEXT i
RETURN

xsend:

PRINT
PRINT "Send what filename";
INPUT file$
IF file$="" THEN abort
ack$=CHR$(6)
block=0
nak$=CHR$(21)
eot$=CHR$(4)
CLOSE 3
OPEN file$ FOR INPUT AS #3
ln=LOF(3)
PRINT "Length of File ";ln; ". ";
n=ln/128
IF INT(n) < n THEN n=INT(n)+1
ELSE n=INT(n)
PRINT n;" Blocks."
bf$=""
80 GOSUB waitchar
IF char$<>nak$ THEN 80
75 IF ln>127 THEN
  bf$=INPUT$(128,3) ELSE GOSUB
getrest
  ln=ln-128
  block=block+1
90 PRINT "  Sending Block";block
  GOSUB stuff
  PRINT #2,bf$;
  GOSUB checksum

```

```

PRINT #2,CHR$(chsum);
GOSUB waitchar
IF char$=nak$ THEN BEEP:PRINT
"Checksum Error in
block";block:GOTO 90
IF char$=ack$ AND ed=0 THEN 75
IF char$=ack$ AND ed=1 THEN
PRINT #2,eot$
IF char$<>nak$ AND char$<>ack$
THEN PRINT "Connection Aborts
Transfer":PRINT
:GOTO term
ed=0
PRINT "File Transfer Complete."
GOTO term

```

waitchar:

```

IF LOC(2)=1 THEN
IF LOC(2)=1 THEN
char$=INPUT$(1,2)
RETURN
END IF
GOTO waitchar

```

getrest:

```

bf$=""
bf$=INPUT$(ln,3)
FOR i=ln+1 TO 128
  bf$=bf$+" "
NEXT i
ed=1
RETURN

```

stuff:

```

PRINT #2,CHR$(1);
PRINT #2,CHR$(block AND 255);
PRINT #2,CHR$((block AND
255)XOR 255);
RETURN

```

checksummer:

```

chsum=0
FOR i=1 TO 131
  chsum=chsum+
    ASC(MID$(bf$,i,1)) AND 255
NEXT i
RETURN

```

duplex:

```

IF dp=1 THEN PRINT :PRINT "Half
Duplex"

```

```

IF dp=0 THEN PRINT :PRINT "Full
Duplex"
RETURN

```

ed:

```

CLS
PRINT
PRINT
PRINT "  Current Macro's"
FOR i=1 TO 10
  PRINT "F";i;":=";mac$(i)
NEXT i
PRINT
INPUT "Edit Which Macro
(0 to Exit)",which
IF which=0 AND change=1
  THEN dump
IF which=0 AND change=0
  THEN term
PRINT
PRINT "Input New Macro to be
used for key F";STR$(which);"."
LINE INPUT mac$(which)
IF mac$(which)=""
  THEN mac$(which)="Undefined"
change=1
GOTO ed

```

dump:

```

CLOSE 3
OPEN "Macro" FOR OUTPUT AS
#3
FOR i=1 TO 10
  PRINT #3,mac$(i)
NEXT i
CLOSE 3
change=0
GOTO term

```

gettem:

```

ON ERROR GOTO macerr
CLOSE #3
OPEN "Macro" FOR INPUT AS #3
FOR i=1 TO 10
  LINE INPUT #3,mac$(i)
NEXT i
CLOSE 3
RETURN

```

macerr:

```

IF ERR=53 THEN
  CLOSE 3
  OPEN "Macro" FOR OUTPUT AS

```

| | | |
|--|--|---|
| <pre> #3 FOR i=1 TO 10 PRINT #3,"Undefined" NEXT i CLOSE 3 END IF MENU ON RESUME gettem macro: g=ASC(b\$) g=g-128 IF mac\$(g)="Undefined" THEN y=1 ELSE y=0 hy=INSTR(mac\$(g),"[RET]") IF hy=0 AND y=0 THEN PRINT #2,mac\$(g); IF hy=0 AND dp=1 AND y=0 THEN PRINT mac\$(g); IF hy<>0 AND y=0 THEN PRINT #2,MID\$(mac\$(g),1,hy-1) IF hy<>0 AND dp=1 AND y=0 THEN PRINT #2,MID\$(mac\$(g),1,hy-1) GOTO 20 review: FOR i=1 TO LEN(buff\$) STEP 4 PRINT MID\$(buff\$,i,4); IF INKEY\$<>" " THEN RETURN NEXT i RETURN info: WINDOW 5,,(70,50)-(540,140),0,-1 PRINT " Info" PRINT PRINT " SuperTerm V1.0 was written by Kelly Kauffman to be used" PRINT "EXCLUSIVELY on the Amiga. The Program was written in " PRINT "Microsoft Basic, and requires an Amiga with 512K." PRINT PRINT " Program & Audio-Visual Copyright 1986" PRINT " Kelly Kauffman" PRINT PRINT "This program may not be sold for ANY PRICE without consent" PRINT "of the author. Please feel free, however, to distribute" </pre> | <pre> INPUT "free copies of this program. Press [RETURN] ";a\$ WINDOW CLOSE (5) WINDOW (1) WINDOW OUTPUT (1) RETURN clearbuf: buff\$="" bf\$="" PRINT PRINT "Buffer Cleared" PRINT RETURN princhr: CLOSE #2 OPEN com\$ AS #2 LEN=1000 FOR uip=1 TO 200:NEXT uip PRINT #2,jr\$; RETURN direct: PRINT PRINT "Directory Pathname [Or Press Return to use Default Pathname]" INPUT path\$ PRINT FILES path\$ RETURN chdf: PRINT PRINT "Enter pathname to be used as the Default Directory" INPUT ty\$ CHDIR ty\$ PRINT RETURN killer: PRINT PRINT "Enter Filename to delete"; INPUT ty\$ PRINT "Are you Sure (y/n)"; INPUT yn\$ IF yn\$<>"Y" AND yn\$<>"y" THEN PRINT "File ---->>NOT<<==== Deleted" RETURN </pre> | <pre> ELSE KILL ty\$ END IF PRINT ty\$;" is gone!!!" PRINT RETURN ren: PRINT PRINT "Rename:" PRINT "Input OLD filename to rename"; INPUT oldfil\$ PRINT PRINT "Input NEW name for the file"; INPUT newfil\$ NAME oldfil\$ AS newfil\$ PRINT PRINT oldfil\$;" is now called ";newfil\$;" " PRINT RETURN breaker: PRINT #2,CHR\$(3); RETURN mem: PRINT PRINT FRE(-1);" Bytes of Memory Remain." PRINT RETURN </pre> |
|--|--|---|

Miga-Mania

by Perry Kivolowitz

Well, here we are, with our second issue of **Amazing Computing™**. As mentioned in last month's Miga-Mania, your opinions, contributions, and suggestions are completely welcome and, in fact, are encouraged. You can make your views known to us by:

USMail: Miga-Mania
c/o Amazing Computing
PiM PUBLICATIONS, Inc.
P.O. Box 869
Fall River, Ma. 02722

Usenet:ihnp4!ptsfa!well!perry

Compuserve:Don Hicks 76714,2404

Please feel free to suggest topics for inclusion in future Miga-Mania columns. If you would like to discuss a past Miga-Mania column, please include the Miga-Mania header (example: General Helpful Hints #1).

This month I have several helpful and interesting things to discuss. First, I would like to devote a paragraph or two in each Miga-Mania column to discussing what is happening at various Amiga user groups around the country. If you know of an Amiga user group in your area or would like to find other Amiga users in your area who would like to start an Amiga user group, drop us a line. This month, I will spend a bit more than a paragraph on the subject of user groups just to get things started. But, as I said, in future issues I will keep to a paragraph or two.

User Group Information: #1

What Can A User Group Do For You?

or

"The User Group Manifesto"

User groups focusing on the Commodore Amiga Personal Computer are, as put by The Amigo in last month's AC, "popping up everywhere." Most are run as non-profit informal groups dedicated to the dissemination of information about the Amiga. Others are run as profit making businesses. The reasons for joining a user group include:

1. User groups, because they represent a large number of potential buyers, can attract vendors of hardware and software to demonstrate their wares, often before their product is available on the open market. Being a smart computer consumer means being familiar with a product before you buy. User group demonstrations allow you to make an informed purchasing decision.

For example, the Jersey Amiga User's Group, mentioned below, had 160 people at the last meeting. The President of Manx Software Systems spoke on the Aztec C Compiler. Also, a technical manager of Activision discussed new releases from Activision.

2. User groups should be able to turn their membership roster into a powerful aid in procuring discounted goods and services from vendors interested in speaking at a group meeting. Because YOU and all the other members of the

group represent a large potential consumerbase, vendors should be willing to make discounts available on either a "discount to those here tonight" or "mention your user group and get a discount" basis.

3. You can look towards your user group as a source of fraternal and mutual support (of the moral and technical kinds). If you have a problem, chances are other members of the group are experiencing the same problem and someone in the group may be able to help.
4. User groups should strive to establish a public domain disk library. The JAUG now has ten packed disks in their library and the number will continue to grow. You could have (extremely) low cost access to priceless examples, tutorials and PD utilities. (Do not forget AC's access to PD software for you and your user group. Ed Note)
5. Many user groups can offer a newsletter containing useful local discussion of Amiga oriented issues. Generally, user group newsletters are around four to eight pages long, perhaps containing discount coupons from stores near you.

While you cannot expect a locally done user group newsletter to contain as much technical information as **Amazing Computing™**, what you can expect, and why such a newsletter is valuable, is information pertaining specifically to YOUR local area. For example: bad (or good) experiences with local dealers.

6. Very active user groups can even institute a series of public training seminars in such areas as Amiga Basic and C programming.
7. User groups should **NOT** allow copying of non-public domain disks at group meetings or as an "informal" part of group activities. Now, I am not expressing any opinion one way or the other concerning copy protection or any of its related issues. I am merely pointing out that user groups are far too valuable a tool for the Amiga using public to lose should the group be forced to disband because they were actively violating the law.

In summary, the old adage "there's safety in numbers" is really the strongest case for joining a user group near you. You will meet many people having similar backgrounds and similar needs. Together, you can use and enjoy your Amiga Personal Computer better than in a vacuum.

To get things rolling, here's information on the Jersey Amiga User's Group:

Name: Jersey Amiga User's Group
Where: Lecture Hall 114, Hill Center
Rutger's University,
New Brunswick Campus
New Jersey

When: Usually third Friday of each month. 7:30 p.m.

Dues: \$15.00 per twelve months.

Newsletter: Planned

PDLibrary: Currently 10 disks, including Fish Disks 1-8.

BBoard: None

Meeting Size: More than one hundred persons per meeting.

Upcoming Speaker: Cherry Lane Technologies

Contact: Perry S. Kivolowitz (201) 271 - 4522. (I can not return long distance calls)

For future Miga-Mania columns, send along information such as the above and it will be published here. As a policy, I will not publish listings from user groups run for profit since appearing in this column is free advertising I want to make sure there's room for any non-profit group that would like exposure. Please include a contact name and phone number so people interested in the group or vendors interested in speaking to the group can get in touch with you.

(Editor's note: Please send in your Amiga user group information and we will include your group in our upcoming user group information issue.)

AmigaDos Bugs #1 --- Epson Driver Overlap Problem

Anyone using the Epson printer driver to print bit mapped images will notice that there seems to be a doubly stricken line every quarter of an inch or so. A small bug in the 1.1 Epson printer driver (note this is NOT a problem with your printer) causes the paper to advance one scan line (row of pixels) too few with each pass of the print head. This causes the top scan line of the current pass to overlap the bottom scan line of the previous pass.

The problem has already been fixed and will be shipped in version 1.2 of the operating system. However, the 1.1 Rom Kernel Manual (RKM) contains the source to the Epson printer driver and the required correction has been circulated by Commodore so expect someone to make the change and place it on a public domain disk probably before the official 1.2 makes it to your local dealer.

Intuition #2 --- The Flood Call Must Have A TmpRas

The C language programmer has access to an Intuition routine called **"Flood."** Flood takes a pointer to a **"rastport,"** a mode flag, and starring x and y coordinate with the rastport. Every window has a rastport but you can also have a rastport independent of any window. Rastports are the data structure which point to the memory where you will be storing the bits comprising some sort of image. For example, a window structure points at a rastport struct which points at the memory where the current image of what is in the window is stored.

Window -----> RastPort -----> BitMaps
(visual content of window)

You tell Flood where to begin its operation by specifying an x and y coordinate with respect to the bit map (the upper left hand corner of the bit map is the origin with x increasing to the right and y increasing downwards).

The mode flag is either 1 or 0. When the mode flag is 0 you are flooding in "outline mode." That is, the region (of pixels within the rastport) bounded by pixels whose color is defined by the "area outline pen" will be set to the color specified by the current "foreground pen."

Mode 1 is "interior mode." In this mode, the current foreground pencolor will be loaded into all horizontally and vertically adjacent pixels whose current color matches the color found in the pixel specified by the x and y coordinate.

Anyway, the point of this note is to make clear to you that Flood calls will not operate correctly without defining and initializing a temporary rastport (or "TmpRas") structure. The TmpRas structure is going to be used as a scratch pad for Flood calls as well as any of the "Area" calls. More information on these calls can be found on page 8-42 of the 1.0 RKM where it is not stated clearly enough that the TmpRas structure will be required by the Flood primitive.

A single plane bit map must be allocated (using AllocRaster) with the same width and height as the rastport with which it will be used. Remember to check the return value from AllocRaster to make sure that it really did return the requested space to you (see Programing Quicky #1, AC Vol. 1, No. 1 for what to do if space was NOT available).

Next, initialize the TmpRas structure by calling InitTmpRas passing it pointers to your TmpRas structure, the bit map you previously allocated, and the size (IN BYTES) of the allocated bit map.

INTRODUCING... THE MICRO-SYSTEMS PERSONAL PRODUCTIVITY SERIES

Analyze! is a powerful electronic spreadsheet program. Essentially, this program is a full-screen calculator where you can organize your data into rows and columns. These rows and columns can then be analyzed with simple mathematics or complicated formulas. Rows and columns can be duplicated to avoid re-typing. Both data and formulas can be edited with only a few keystrokes.

From home budgets and check registers to financial modeling and your company's general ledger, all manner of bookkeeping tasks become faster and easier with Analyze!. An astounding value at only \$99.95!

OnLine! combines features and convenience in a high quality package that will meet all of your telecommunications needs. With OnLine!, you can use your Amiga as a window to the world of information that is just on the other side of your telephone. You can link up with commercial information services for stock quotes, airline information and reservations, technical databases, and local bulletin board systems (BBS) and discover a new world of information and software for your computer. Corporate users can use OnLine! to let their Amiga access data stored on the company's mainframe computer.

OnLine! is the finest program of its type available for the Commodore Amiga. You can't lose when you get "online" with OnLine!. All for a down to earth price of only \$69.95!

Scribble! puts word processing efficiency at your fingertips. If you write anything from grocery lists to novels, you need this program. This is a word processor that helps you get your writing job done without needing a degree in physics to operate it. Scribble! is carefully designed with simple controls that make SENSE (unlike some other programs, which require you to take a week's vacation from work in order to learn how to use them). When you purchase Scribble!, you'll be using the program successfully only minutes later. Now THAT'S value for your dollar!

Despite its simplicity and whimsical name, Scribble! is a full featured word processor. Advanced formatting commands are included to give you a great deal of control over how your printed document will look. The editor will use any normal ASCII file (no rejection of carriage returns here!). It features multiple windows (up to four documents being edited at the same time) with block transfer between windows. And, through it all, Scribble! uses the famous Amiga Intuition interface. Menus are controlled with the mouse and all windows can be "sized" and moved as the user desires. Scribble! can even be printing out a document while you're using your Amiga for something else! Only \$69.95.

Micro-Systems Software, Inc.
4301-18 Oak Circle
Boca Raton, FL 33431

(800) 327-8724/National, (305) 391-5077/Florida

Ask about our Amiga bulletin board program, BBS-PC!. AMIGA, OnLine!, Analyze!, Scribble!, and BBS-PC! are trademarks of Commodore-Amiga, Inc., and Micro-Systems Software, Inc., respectively.

Finally, associate the initialized TmpRas structure with your rastport by placing the address of the TmpRas structure into the TmpRas field of the rastport structure.

When you are done, remember to deallocate the bit map you allocated with the call to AllocRaster.

CLI Quicky #1 --- Searching All The Files In A Directory For A Pattern

Remember in issue one of Amazing Computing I said I did not have an AmigaDos User's Manual (AUM)? Well, I finally got one and boy, many of the complaints I had about the CLI environment have been dispelled! For example, I needed a "grep-like" capability time and time again. Little did I know that **SEARCH** was exactly what I needed.

Since George Musser, Jr. will be going over the **SEARCH** command as part of his detailed introduction to the CLI, I will be brief here, providing enough for you to be able to start **SEARCHing** right away.

The "**SEARCH**" command is really a very powerful utility. Very often, there is information that you need to get from one or more of the files on a disk but you

can not quite remember which files to look in. Using **SEARCH** you can specify a pattern to be sought and have the Amiga automatically search all the files in a given directory or even all the files on a given disk!

You use **SEARCH** in one of the following ways:
SEARCH <in-what-directory> <for-what-pattern>
or
SEARCH <what-disk-drive> <for-what-pattern>

In both cases the specified pattern will be sought for in all of the files in the named directory. If you specify a disk drive name (such as "df0:") all of the files in the root directory of that device will be searched.

You can follow each of the above commands with the word, "**ALL**" which will cause **SEARCH** to examine all the files in every subdirectory of the named directory. Thus, if you wanted to find every file on an entire disk that had the pattern "tax deductible" in it, you would say:

SEARCH df1:"tax deductible" **ALL**

Notice the double quotes around the pattern. They are needed because the pattern has a space imbedded in it. The double quotes make sure that the **SEARCH** command sees the entire pattern as one string rather than as two distinct arguments.

Another nice feature is that you can also specify a pattern for the names of files to be searched. For example:

SEARCH df1:IRS-1099-8#? **WAGES**

If you stored all your tax forms for the last forty years on the diskette in drive 1 using the scheme:

"IRS-<Form Number>-<Last Two Digits Of Year>

The above command will look for the word "wages" (letter case does not matter to **SeArCh**!) in all of your 1099's filed in the 1980's.

Each time **SEARCH** finds an instance of the sought after pattern, the file currently being searched as well as the line containing the pattern will be printed. You can interrupt the search at any time by depressing the control key along with the letter "c."

For a more in-depth look at **SEARCH** refer to George Musser, Jr's series on using the CLI.

That will be all for this month. I hope to bring you more Miga-Mania next month in the third issue of Amazing Computing. We're new. We're hungry. We want your feed back. No, that's not right. We want your feed bag! No, that isn't it either. What we're trying to say is:
WE WANT YOUR FEEDBACK!

•AC•

AMIGA OUTLET

| | | |
|---|------------|----------|
| 3 1/2" Disks (DS, DD) | 10/\$29.95 | 1/\$3.15 |
| 3 1/2" Disks (DS, DD) Plain Label Brand | | \$call |

CLASSIC IMAGE, INC. - PRESENTS
DIABLO - Graphic mind challenge game \$29.95
DISK LIBRARY - Now you can File, Catalog, Update Search, Cross Reference, Report \$49.95

Amiga System Covers - W/mouse/LOGO \$21.95
Amiga Disk Cover - 1010 or 1020 with LOGO \$7.99
Paper T/F-F/F White, 9 1/2 x 11, 20lb. 150/\$8.99
Paper T/F-F/F White, 9 1/2 x 11, 20lb. 1000/\$22.95
Paper T/F-F/F 1/2"Greenbar, 9 1/2 x 11, 18lb1000/\$18.99
Index Cards - T/F-F/F, 3 x 5 \$500/\$7.95
Rolodex Cards - T/F-F/F, 2 1/6 x 4 \$500/\$8.95
Labels - T/F-F/F, Address 1000/\$5.00

S&H-\$2.50 US M.W. RUTH CO., AMA36
S&H-\$4.50 CN Visa 510 Rhode Island Ave.
US \$'s only Master Cherry Hill, NJ 08002
(609) 667-2526

We stock what we sell, for fast delivery.

Send for **FREE CATALOG**•All available **AMIGA** items

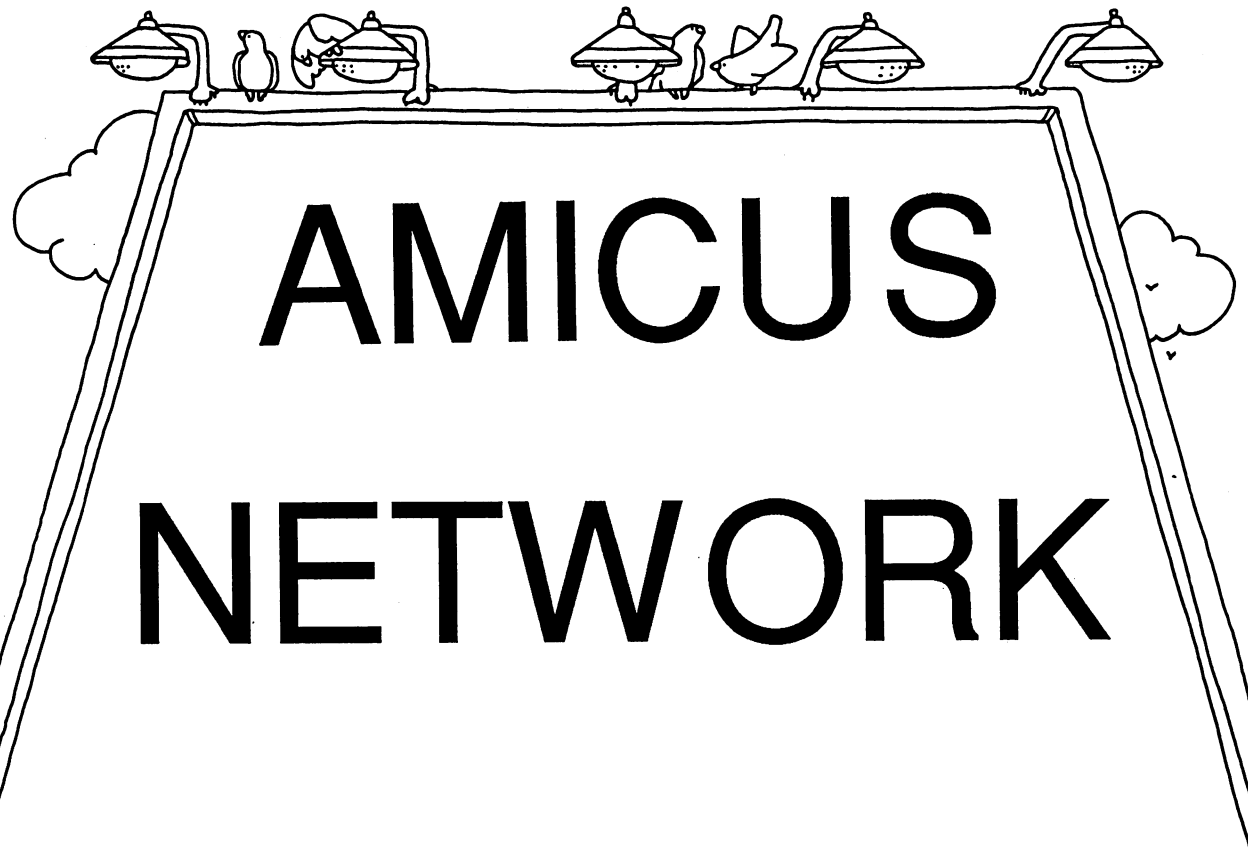
ATTENTION PROGRAMMERS - Let us take over the headaches of publishing your software. We are looking for all items related to the "AMIGA".

AUG

AMIGA USERS' GROUP 68000

You will receive our official newsletter. Evaluations on software and hardware. Advanced updatings, technical information, Problem-solving, program exchange, Buying discount service, and much more. Send \$18.00 US for Charter Membership to:

AMIGA USERS' GROUP 68000
Box 3761 - Attn: Jay Forman
Cherry Hill, NJ 08034
(609) 667-2526 • Visa/Master-Add \$1.00



AMICUS NETWORK





The AMICUS



by John Foust



What is the AMICUS Network?

If you missed the premiere issue of *Amazing Computing* magazine, you missed the explanation of the AMICUS Network. In summary, AMICUS is a group for Amiga users and developers. It started in the summer of 1985 to bring Amiga developers together on the Usenet, but soon expanded to include developers and users everywhere.

The Amicus Network forms an umbrella group over all Amiga users and user groups. It does not collect dues or dictate policy. Yet, we have collected several disks of public domain software, available through this magazine, or from your local user group and we welcome local user groups to contribute their latest software.

This regular section of *Amazing Computing* reports the affairs of the AMICUS Network. Expect to see articles of a technical nature in this section. A large percentage of AMICUS members are developers, and everyone is encouraged to contribute articles on Amiga programming style.

Any Amiga user or user group is welcome to distribute the AMICUS disks. Of course, we hope that someone will not tarnish this volunteer effort by trying to sell, at ridiculous prices, the software we have gathered.

Do not be taken in by advertisements for public domain software. Chances are, the same software is available somewhere else for a lower cost, including free. The cost of a full disk of programs, duplicating, and postage is surely less than ten dollars.

Public domain disks

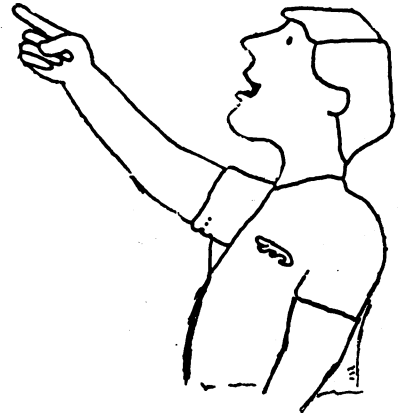
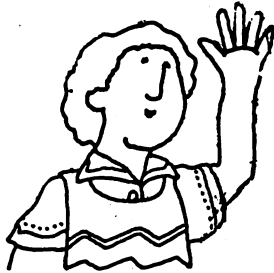
I expect the size of the public domain library to double or triple in the next month. By the time you get this, I will add many exciting programs mentioned in last month's column. Don't forget that all programs published in *Amazing Computing*™ will join the public domain disks.

The programs currently on disk include: XLISP, a version of Lisp; Hack, an adventure game; a version of the Icon programming language; MVP-FORTH, a shareware version of FORTH; a Star Trek game; several more text formatters; a version of Unix 'compress', a program to shrink file storage space; several banner-printing programs, plus many more examples of Amiga code, in ABasic, Amiga Basic, C, assembler, Pascal, and Lisp.

A lot of public domain C code has been coming from Fred Fish of Pleasant Hill, California. Fish has collected over 300 megabytes of programs from his work on Unix and microcomputer systems. Slowly but surely, many people are converting this software to run on the Amiga. He expects to make several disks of software available to Amiga users each month. *Amazing Computing* will offer the disks, as available through our regular distribution of Public Domain Software.

Commodore Canada has offered to introduce AMICUS to TPUG, the Toronto Pet User Group, one of the largest user groups in existence. TPUG has collected several disks of public domain software from their Amiga special interest group. The TPUG software should be in the library soon.

NETWORK



Now, if we could persuade Commodore Amiga to release all those basement programming tools and demos to the public domain, and find some oldie-but-goodie TRIPOS utilities ...

IFF and graphics

The IFF data interchange format is detailed in the version 1.1 developer's manuals. This format specification is public domain, to encourage a similar data format for all Amiga programs, so the results of one program can be used in another. AMICUS Volume 2 has a subdirectory with the necessary IFF files for C programs. However, this version is not the official version, since the programmers at Electronic Arts are still fine-tuning the IFF standard.

The IFF examples include a program that views any file in the IFF format. Since Electronic Arts' Deluxe Paint uses IFF, this means people who do not own Deluxe Paint can still view the creations of others. I hope future AMICUS disks will include a gallery of fine paintings and brush assortments.

An extension of this IFF display program was posted to CompuServe. The program displays a given series of IFF files, much like a programmable slide viewer.

Draper tutorials

John Draper, one of the first employees of Apple Computer, has written several extensive tutorials on Intuition menus, requesters, gadgets and animation. They can be found on AMICUS PDS Volume 2.

These texts serve as explanations of the, sometimes obscure, developer's manuals. They include code examples of a much more general nature than the examples given in the official manuals.

For example, his gadget tutorial explains each type of gadget - boolean, proportional, string and integer - in both prose and C code. It includes a C program that demonstrates the use of each.

A working example provides a deeper level of confidence than the version 1.0 developer manuals. Written under the deadline pressure, the version 1.0 manuals have a few bugs. Version 1.0 was not really in place when some of the sections were written. It is hard to trust manuals written with a different version of the operating system. It is also much easier to use Draper's examples, compared to typing in the examples in the developer's manuals.

Some of you might recognize Draper as Captain Crunch, the phone phreak who helped Steve Jobs and Steve Wozniak in their first business venture, making the little boxes that made free phone calls. Draper has posted his tutorials to the Well, a California-based programmers network, which has insured their distribution on Usenet and many other networks.

Amiga Link

Several months ago, Amiga developers were mailed an offer to join Amiga Link, a national Amiga technical support network that replaces the San Jose Perennial Support Service computer, which closed on December 20.

In the third week of January 1986, Amiga Link opened for business. The network is staffed by technical support people from Commodore-Amiga. So far, the staff has quickly answered most questions in a manner that puts the old tech support BBS to shame.

In December, I posted a question on the San Jose computer, and it was never answered, even though I followed it up with a (voice) phone call. The programmer I spoke with promised to post the answer and send some additional documentation in the mail, but nothing ever arrived. I posted this question as the first question on Amiga Link, and it was answered on the next business day, with a C code example.

Announcing

Amiga Project

Programming Journal For The Amiga

A no—nonsense journal dedicated to software development for the Amiga on the Amiga.

| | |
|---------------------|----------------|
| Subscribe to | Amiga Project |
| Amiga Project | P.O. Box 285 |
| 12 issues...\$24.00 | Kent, OH 44240 |
| | 216-673-0185 |

Dealers, Advertisers call 216-673-0185

Workbench 'more'

The Amiga Link question was about the information passed to a program when the program is started. If the program is started from the CLI, the command line arguments are passed in `argc` and `argv[]`, in the standard C fashion. If the program's icon is double-clicked from the Workbench, some message structures are initialized, which present similar information, and more.

This information is necessary to write programs such as a Workbench-based 'more' program. 'More' is a Unix program to display text files. This kind of program is a virtual necessity for the AMICUS disks. It is hard to tell novice users that they have to use the CLI to view text files, since the explanation involves a lesson on subdirectories and everything else the Workbench avoids.

The Workbench 'more' program source code is described in Rick Wirth's article elsewhere in this issue, and the executable version can be found on all AMICUS disks.

A possible extension of this program could launch ABasic programs from the Workbench. Again, this program will make the AMICUS disks nicer for novice users, since they will not have to learn the ABasic 'chdir' command. The program would rename a given program to 'init.bas', start ABasic as a process, and rename the file back to its original name a few seconds later. ABasic will load and execute any program named 'init.bas' when it starts.

AMICUS membership

With the advent of this column, the meaning of 'AMICUS membership' has changed. At first, I planned to write a newsletter, and send it to AMICUS members. Instead, I sent a copy of the premiere issue of Amazing Computing to all original members of AMICUS, in lieu of a newsletter.

Does AMICUS membership mean anything, then? Well, the first hundred people got a complimentary copy of Amazing Computing. People who made substantial donations will get another, if all goes well. That is about it for now. I have no doubts that a subscription to Amazing Computing will be of greater benefit than a newsletter.

A possible method of funding AMICUS is to maintain and sell a mailing list of known Amiga owners. I am sure many Amiga developers would purchase this list in computer readable form, and many Amiga users might want to see offers of new Amiga software. User groups could contribute a list of their members who desire to be on a list of this sort. If this sounds like a good idea, please write, or suggest other methods AMICUS can benefit Amiga owners.

User groups

If your Amiga user group wants the AMICUS disks, please appoint one person as the AMICUS liason, and send a request for the Public Domain Disks to:

PDS Software

Amazing Computing

PiM Publications, Inc.

P.O.Box 869

Fall River, MA 02722

Later, if your group gathers new public domain software, send it to me at the address below, with a stamped return mailer. I will add your contribution to the next volume of the library as long as it is not currently in the library. As a bonus, I will make a copy of one or two of the latest AMICUS disks, depending on the number of disks you send. In this fashion, your group gets the latest AMICUS software, AMICUS gets your latest contributions, and I, as well as my disk drives, get a break.

Call for writers

If you have an interesting programming tip or technique, please consider writing an article for Amazing Computing. The Amiga programming environment still has a few quirks that only heavy programming projects will encounter. Your explanation will bring fame and fortune, no doubt.

AMICUS address

If you are interested in writing for the AMICUS section of Amazing Computing, please contact me at CompuServe 72237,135, Source BDN388, PeopleLink INQ740, Amiga Link "John Foust"; or at: Sound Creation, Ltd., Attn: John Foust

900 John Nolen Drive

Suite 130-B

Madison, WI 53713.

Please do not call.

•AC•

UBZ FORTH™

for the Amiga™

- *FORTH-83 compatible
- *32 bit stack
- *Multi-tasking
- *Separate headers
- *Full screen editor
- *Assembler
- *Amiga DOS support
- *Intuition support
- *ROM kernel support
- *Graphics and sound support
- *Complete documentation
- *Assembler source code included
- *Monthly newsletter

\$85

Shipping Included
in continental U.S.
(Ga. residents add sales tax)

UBZ *Software*

(404)-948-4654

(call anytime)

or send check or money order to:

UBZ *Software*

395 St. Albans Court
Mableton, Ga 30059

Try FORTH Be Impressed

*Amiga is a trademark for
Commodore Computer. UBZ FORTH
is a trademark for UBZ Software.

A Workbench "MORE" PROGRAM

by Rick Wirch

(More is a program to view text files from the Workbench. This program, in executable form, is on each of the AMICUS disks.)

Amiga users tend to first learn to use the Intuition interface, and then graduate to the CLI. Most programs for other computers depend on a similar command line interface, and the standard C method of passing that command line, `argc` and `argv[]`. The question is, what is the easiest way to make an existing program work with the icon interface?

Any program started by double-clicking an icon on the Workbench can access information similar to `argc` and `argv[]`.

In the example program given here, the code above the dashed line converts Workbench information to the standard C `argc` and `argv[]`. This code can be lifted intact, and ported to make old-style programs interact with the Workbench. It allows invocation from the Workbench to be completely transparent to programs that normally use command line filename arguments.

The remaining code listing is a simple 'more' program, based on the Unix program to display text files.

The key to dealing with the Workbench, lies in an external pointer to a `WBStartup` struct called `WBenchMsg`. This pointer is initialized by Intuition when a program icon is double-clicked. It is valid when `argc` is zero. `Argc` is always non-zero when called from CLI, since the first element is the program name.

First, `argc` or argument count is set to the value in `WBenchMsg->sm_NumArgs`. This value is the Workbench equivalent of `argc`.

Second, a pointer is initialized to the list of `WBArg` structures. This list has as many elements as `WBenchMsg->sm_NumArgs`, and each element contains a pointer to a string called '`wa_Name`', and a `FileLock` pointer called '`wa_Lock`'.

The string '`wa_Name`' contains the name of the argument (usually a filename) and a **FileLock pointer** to the argument file. Note that the file name does not include the path name.

This program allocates new memory for each argument, copies each argument name to the new space, and then sets each `argv[]` element to point at the string. The boolean variable '`CLI`' is set so the program can free this memory allocated before terminating the program.

Since the file's leading path name is not given, the program changes to its directory using the **CurrentDir()** function. This must be done before attempting to **Open()** it. It is good programming style to change back to the initial directory at the completion of the program.

There are two different ways to use this program. Method one- create icons for each text file you want to view. From the Workbench, single-click this icon, and using the Workbench Info menu item, set the default tool of each icon to the name of this 'more' program. When this text icon is double-clicked, the 'more' program will run, displaying this file.

Method two - create an icon for this 'more' program, copy the Clock icon and draw a new picture for it. Any text file you wish to view must have an icon of any type. Single-click the text icon, and then hold the shift key down while you double-click the 'more' icon. The 'more' program will run, and the name of the text file will be passed as an argument. This shift-click method is known as **extended selection**.

This program will view NotePad files without much trouble, but Textcraft complete document files contain a lot of binary, so the results are less useable.

If you try to select program icons, like the Clock or Calculator, and then extended double-click the 'more' icon, the Clock and Calculator programs will run, but the 'more' program does not. I don't know why, either.

This technique should open doors for many existing programs.

MORE listing:

```
/* a Workbench-based 'more' program */
/* by Rick Wirth */
```

```
#include "exec/types.h"
#include "exec/memory.h"
#include "libraries/dos.h"
#include "libraries/dosextns.h"
#include "workbench/startup.h"
```

```
#define BRK 0x03
#define ESC 0x1b
```

```
extern struct WBStartup *WBenchMsg;
extern struct FileHandle *Open();
extern struct FileLock *CurrentDir();
```

```
main(argc, argv)
```

```
int argc;
unsigned char *argv[];
{
    struct WBArg *arg; /* pointer to argument list
    struct */
    struct FileLock *oldlock; /* lock to old directory */
    struct FileHandle *window = NULL; /* file handle for
    raw window to screen */
    struct FileHandle *workfp = NULL; /* file handle to file
    to be displayed */
```

```
    register short int i, len, len2; /* loop and counter
    variables */
    short int linecount = 1; /* count of lines displayed
    to screen */
    short int CLI, done; /* boolean variables */
```

```
    unsigned char c; /* character read from the
    keyboard */
    unsigned char work[80]; /* string space used for
    messages */
    unsigned char buf[512]; /* buffer to hold one
    block of the file */
```

```
    CLI = TRUE;
```

```
    /* if the program was invoked from Workbench, argc
    is zero. */
```

```
    if (argc == 0) {
        CLI = FALSE;
        /* change argc to WB arg count. Set pointer to
        argument list */
        argc = WBenchMsg->sm_NumArgs;
```

```
        arg = WBenchMsg->sm_ArgList;
        /* Point the standard argv[] pointers at the
        Workbench arguments */
        for (i=0; i < argc; i++, arg++) {
            argv[i] =
                (unsigned char *)AllocMem(strlen(arg-
                >wa_Name), MEMF_PUBLIC);
            strcpy(argv[i], arg->wa_Name);
            /* Get a lock on the directory where the file
            resides */
            if (i == 1)
                oldlock = CurrentDir(arg->wa_Lock);
        }
    }
```

```
    /* -----end of argument handling---
    ----- */
```

```
    done = FALSE; /* boolean for short circuit
    ending */
```

```
    /* Try to open the file to display. If there is an error,
    stop. */
    workfp = Open(argv[1], MODE_OLDFILE);
    if (workfp == NULL)
        done = TRUE;
    else {
        /* Open a Workbench window. If there is an error,
        stop. */
        strcpy(work, "Raw:0/0/640/199/ Listing of ");
        strcat(work, argv[1]);
        strcat(work, "");
        window = Open(work, MODE_NEWFILE);
        if (window == NULL)
            done = TRUE;
    }
}
```

```

/* This version of 'more' can only display one file. */
if (argc != 2) {
    if (argc == 1)

        strcpy( work, "Usage: more <filename>");
    else
        strcpy( work, "This program only displays one
            file.\n");
    Write( window, work, strlen(work));
    strcpy( work, "Press SPACE to continue.");
    Write( window, work, strlen(work));
    while (Read(window, &c, 1), c != ' ')
        ;
    done = TRUE;
}

if ( !done)
do {
    /* Read a block of the file, search for end of line. */
    len = Read(workfp, buf, 512);
    len2 = 0;
    while (len2 < len) {
        for ( i=len2; buf[i] != '\n' && i < 511; i++)
            ;
    }
}

```

```

Write( window, &buf[len2], i-len2+1);
len2 = i + 1;
if (++linecount >= 22 && buf[i] == '\n') {
    linecount = 1;
    /* print the messages in color 3 */
    strcpy( work, "\x1b[33mPress SPACE for
        more...\x1b[0m");
    Write( window, work, strlen(work));
    /* wait for a key */
    while( Read(window, &c, 1), c != ' ')
        if ( c == BRK || c == ESC) {
            len = 0; /* short circuit the loop */
            break;
        }
    /* RETURN scrolls by one line */
    else if ( c == '\r') {
        linecount = 21;
        break;
    }
}
/* erase the message */

strcpy( work, "\r          \r");
Write( window, work, strlen(work));
}
}
} while (len == 512);

```

HARVSOFT

PRESENTS

INFO BASE

Now available for the Amiga!

INFO BASE is a powerful, yet easy to use Data Base program, designed for the Amiga to store and retrieve information in an organized manner. Each record can contain up to 200 fields of information.

You create, design, and edit custom print forms. Each print form, you design, can create a different type of printout such as mailing labels, multi column reports, invoices, lists, etc.... In addition, text can be incorporated into the design of print forms, allowing both text and field information to be printed on the same line.

INFO BASE

FEATURES

- *DESIGN DATA BASES, ALLOWING UP TO 200 FIELDS OF INFORMATION FOR EACH RECORD
- *SORT RECORDS BY ANY FIELD
- *SELECT STARTING AND ENDING SORT INFORMATION
- *PRINT OUTPUT TO PRINTER OR SCREEN
- *PRINT MULTIPLE COPIES
- *CREATE, DESIGN, AND EDIT CUSTOM PRINT FORMS
- *COMBINE TEXT INTO THE DESIGN OF PRINT FORMS
- *ADD RECORDS AND DELETE RECORDS
- *BROWSE THROUGH RECORDS
- *SEARCH FOR SPECIFIC RECORDS

INFO BASE is available NOW!

\$45.00 (US)

Enclose Check, Money Order, or Cashiers Check.
Phone orders will be shipped C.O.D. and must add \$3.00 S & H.
Please allow 10-14 days for delivery.

Harvsoft

Box, 725
Kenmore, NY 14217
(716) 877-3510

Dealer Inquiries Welcome

INFO BASE requires an Amiga with 512K RAM and at least one disk drive.

```

/* If not aborting the program */
if (!done) {
    strcpy( work,
        "\n\x1b[33mEnd of File. Press
SPACE\x1b[0m");
    Write( window, work, strlen(work));
    /* wait for spacebar to quit */
    while (Read(window, &c, 1), c != ' ')
        ;
}

/* if NOT spawned from CLI, free memory, return to
last directory */
if (!CLI) {
    for (i=0; i<argc; i++)
        FreeMem( argv[i], strlen( argv[i]));
    oldlock = CurrentDir( oldlock);
}
if (workfp)
    Close(workfp);
if (window)
    Close(window);
exit(0);
}

```

Amiga BBS Numbers:

Special thanks to Richard Rae, CIS 72177,3516 and
Jason Goldberg, CIS 72737,1070

| Phone | Name | Notes | | | |
|--------------|---|---|--------------|---|--------------------------|
| | | | 415-581-9452 | Sphinx | 1200 B, CP/M |
| 201-446-1424 | Amy Palace | 1200 B | | | |
| 203-234-0742 | Firesign Theater | 1200 B Fidonet | 415-595-5452 | FAUG 6 pm-7am 24 hrs on weekends | 300 B, |
| 203-234-0742 | Lido Hotel | 1200 B Fidonet | 415-659-9169 | RSVP | 1200 B Fidonet |
| 206-874-6381 | StoneHenge | 1200 B, 5 pm to 8am 24 hrs on weekends | 415-664-2811 | Welchnet | 1200 B Fidonet |
| 212-269-4879 | Amuse | 2400 B Fidonet | 501-988-2064 | MegaHacker | |
| | | | 506-357-9660 | Canada System | |
| 212-927-6919 | 68000 BBS | 1200 B | 513-898-0702 | AmigaBoard | 1200 B |
| 213-478-9788 | Glass BBS | 2400 B | 516-661-4881 | Amy Advantage | 1200 B |
| 213-662-2424 | Hals BBS | 300 B | 602-846-3901 | Amiga Talk BBS | 300 B |
| 301-254-8078 | Doppleganger | 2400 B | 615-868-7860 | Connection | 1200 |
| 303-737-2294 | Wilmington | | 618-378-2133 | Wovsed 200 5 pm to 8 am username 'guest' | 300 B, |
| 303-752-0247 | Amy BBS | 2400 B | 714-493-6094 | Vivid Express | 1200 B |
| 303-988-8155 | Unknown TBBS | 1200 B | 714-660-1462 | Action | 300 B RBBS |
| 305-486-7021 | Amiga Mail | 9pm,7am | | | |
| 305-737-1590 | Micro Systems Sftware | 2400 B | 801-264-8290 | TechniSoft | 1200 B |
| 313-348-4479 | Novi Download voice first @ 348-4477 | 1200 B | 816-474-1052 | Mebbsie | 1200 B |
| 402-291-8053 | Wind Dragon | 1200 B Fidonet | 904-733-4515 | Casa Mi Amiga | 1200 B Fidonet |
| 408-353-4531 | Ace Connection | 2400 B | 914-442-4006 | MusicNet | MIDI board, \$75/year |
| 408-372-1722 | Developers Exchange | 1200 B MaxiCorp | | | |
| 415-332-6106 | The Well | 1200 B VAX | | | |

If you, your company, your store, your user group, or all
three have an active Amiga Bulletin Board System,
contact: BBS
PiM Publications
P.O. Box 869
Fall River, MA 02722

Public Domain Programs

Amazing Computing is proud to offer several disks of public domain Amiga software.

This software is collected from user groups and electronic bulletin boards around the nation, under the auspices of the AMICUS Network, a clearinghouse for public domain software and technical information about the Amiga. Each disk is full of good useful programs.

When source code is provided for any program, the executable version is also present. This means you will not need the C compiler to run these programs.

Most programs and text files have icons visible from the Workbench, however, some files must be accessed from the CLI.

Volume 1:

ABasic programs:

Graphics:

- 3-d drawing program
 - shuttle drawing
 - fractal landscapes
 - painting programs, and more.

Tools:

- address book & card file programs
- speech utility.

Games:

- brickout
- saucer
- othello
- spelling.

Sounds:

- Entertainer
- HAL 900
- Sugar Plum Fairy.

C source code:

- Aterm terminal program
- 'cc'
- color demos
- floating point demos
- fix disks with future dates
- IBM2Amiga transfer
- 'grep'
- Mandelbrot program
- hold-and-modify graphics demo
- wildcard expansion,
- interlace mode on/off
- FreeDraw, a Workbench drawing program
- GfxMem, a graphic memory display
- Dotty demo
- fonts demo, and more.

Executable programs:

- ATerm
- FixDate
- FreeDraw
- GfxMem
- 'grep'
- Mandel
- SetLace
- Sparks,
- Moire
- SpeechToy
 - a speech utility
- fonts demo.

Texts:

- filenames
- 'ed' commands
- half-bright mode
- ROMWack reference
- modem pinout, and more.

Volume 2:

ABasic programs:

- EZ TERM** form AC #1
- Super Spheres** form AC #1

C source code:

- 'ar' file archiver
- 'alib' object library manager
- YachtC, the game,
- sq and usq, file squeezing programs
- simple CLI shell

Assembler code:

- bsearch.asm
- qsort.asm
- setjmp.asm
- System 5 printf.asm
- trees.o

Executable programs:

- 'make'
- YachtC
- showilbm
- an IFF display program
- MicroEmacs editor,
- ar archiver
- alib
- sq
- usq
- simple CLI shell

Volume 2 continued:

- Make: Source code to 'make', a programming utility.
- IFF: Source code and .h files for the IFF graphic interchange format, plus programs to convert Graphicraft to IFF, and saving any screen as IFF.
- Tutorials:
 - John Draper's texts on programming menus, animation, and gadgets.
- Emacs: Source code to the portable MicroEmacs text editor.

Volume 3: (Under Construction!)

- C source code:
 - skewb
 - a -Rubik cube demo program
 - sound demo
 - 6 bit color example.
 - Arrange: 'arrange', a text formatting program, with documentation.
- Executable programs:
 - skewb
 - arrange
 - sound demo.
- Amiga Basic
 - Superterm** form AC #2
- ABasic
 - Patcher

(to be completed sometime in March 1986...)

To receive your copies of the above Public Domain Disks, please send \$7.00 per disk (\$6.00 per disk for Amazing Computing subscribers) to:

PDS Disks
PiM Publications
P.O. Box 869
Fall River, MA 02722

You may enclose your subscription request in the same envelope.

Amazing Computing encourages you to copy these disks in your user's groups and with your fellow Amiga owners. Public Domain is for everyone.

•AC•

INTERACTIVE ANALYTIC NODE

THE EXPLORER

A debug monitor for the AMIGA

THE EXPLORER is a learning tool for new AMIGA owners and a debug tool for the program developer. THE EXPLORER is a machine language monitor similar to the ones you have used on 8-bit machines, but it has some design enhancements that make it more useful. Features: display memory and files in Hex and ASCII, memory modify, search, move, fill, display and change registers, disassembly trace, load programs, disassemble to disk. Output to printer or disk file. Powerful commands: loops, text display, real-time RAM view, & more!

The EXPLORER doesn't force you to enter long commands to perform a function. The command set is compact and efficient. You can execute your commands within loops, creating live displays of RAM or registers while you test your program. You can control the display format too, and even display informative messages. No need to spend ten minutes typing that special data structure into memory every time you debug. Just write a new command to do it for you. After all, what are computers for? When you want to save the contents of RAM or a series of trace steps for future examination, just send them to the printer, or better yet, send them to a disk file!

PRICE \$35 plus \$3 S&H. COD add \$4. Visa/MC orders call (612) 871-6283. Money orders or checks to:

Interactive Analytic Node
2345 West Medicine Lake Drive
Minneapolis Minnesota 55441

Amazing Computing™

Index of Advertisers

| | | | |
|-------------------------------|-------|------------------------------|----------------|
| Advanced Systems Design Group | 19,29 | Lattice, Inc. | 5 |
| Akron Systems Development | 12,40 | Metadyme, Inc. | 2 |
| Amiga Project | 56 | Micro-Systems Software, Inc. | 16,51 |
| Amiga Users' Group 68000 | 52 | Microsmiths, Inc. | C IV |
| Cardinal Software | 33 | M. W. Ruth Co. | 52 |
| Discovery Software | C III | PiM PUBLICATIONS, Inc. | 18,26,30,35,41 |
| Elcom Software | 40 | UBZ Software | 28,57 |
| Harvsoft | 22,60 | ZOXSO | 11 |
| Interactive Analytic Node | 23,63 | | |

COMING NEXT MONTH!

in

Amazing Computing

A continuation of the three Amazing tutorials:

The Amazing C Tutorial

Lisp Tutorial

Inside CLI

The Amazing AmigaBasic™ tutorial part one

Business with Amiga, a new column

ROOMERS

Miga-Maniac.

The AMICUS™ Network

New software and hardware releases for the Amiga™.

New Amiga™ public domain software

and, as always, a few surprises!

BECAUSE YOU'RE GOING TO BUY ONLY ONE
DISK BACKUP UTILITY FOR YOUR



MAKE IT

MARAUDER[®]

MARAUDER IS EVERYTHING YOU EXPECT

Marauder works on any standard Amiga with 256K of memory and one disk drive.

Marauder works without any hardware modification.

Marauder backs up most copy-protected software titles.

Marauder's user interface is clear-cut, easy to understand, and easy to use.

Marauder's manual is written in plain English without confusing technical terms.

Marauder insures that your valuable software investment will be protected
against theft, loss and accidental erasure.

MARAUDER IS EVERYTHING YOU NEED

Marauder takes advantage of Amiga's optional external disk drive and extra RAM if you have them.

Marauder, itself, is not copy protected in any way.

Marauder uses Amiga's advanced graphics and processing power to its fullest advantage.

Marauder automatically formats your backup disk while copying.

Many complicated protection schemes are easily handled by Marauder's
parameter-entry mode — two numbers is all it takes.

Current parameters for all titles will be available **FREE** to registered users at any time.

BEST OF ALL

MARAUDER is AVAILABLE NOW!!!

You can get Marauder at the
SPECIAL INTRODUCTORY PRICE OF



ONLY \$39.95



call now (215) 546-1533

(Dealer Inquiries Welcome)



262 South 15th Street • Suite 300 • Philadelphia, PA 19102-3801

AMIGA USERS:

FIRE YOUR EDITOR.

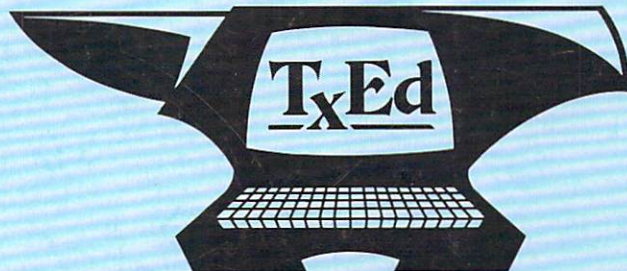
And put Microsmiths' TxEd to use for you.

The Text Editor that should have come with your Amiga is now available.

- Uses mouse and menus
- Fast display and compact code
- Visible highlighting of cut and paste
- Multiple windows
- Edits IBM text files
- Easy to learn, use menus for on line help!

Designed by C. Heath of Microsmiths to meet the needs of programmers. Find out why four out of five Amiga programmers prefer TxEd over any other editor available for the Amiga, regardless of price.

To order, send \$39.95 in check or money order plus \$2.50 postage and handling to: Microsmiths' TxEd, P.O. Box 561, Cambridge, MA 02140. Tel.: (617) 576-2878. Massachusetts residents add 5% sales tax. Amiga is a trademark of Commodore-Amiga, Inc. Dealer inquiries invited.



MICROSMITHS, INC.